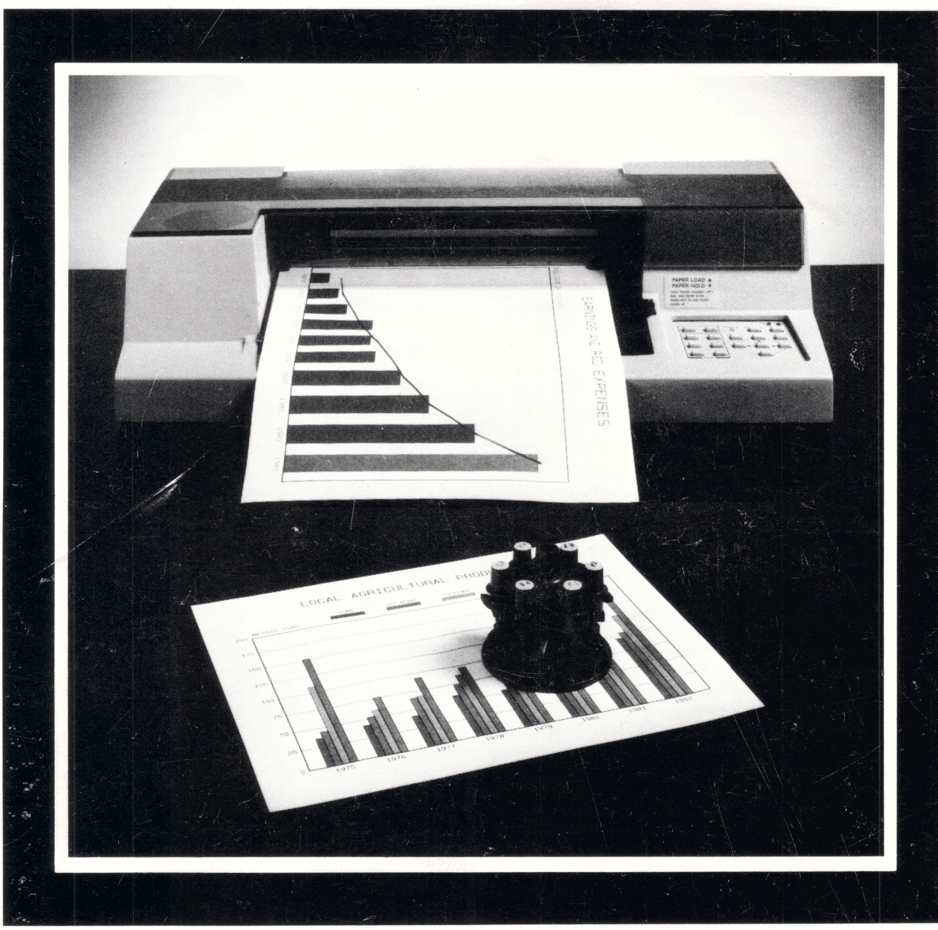
INTERFACING AND PROGRAMMING MANUAL

HP 7475A
Graphics Plotter



The United States Federal Communications Commission (in 47 CFR 15.838) has specified that the following notice be brought to the attention of users of this product.

FEDERAL COMMUNICATIONS COMMISSION RADIO FREQUENCY INTERFERENCE STATEMENT

"This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

— reorient the receiving antenna
— relocate the computer with respect to the receiver
— move the computer away from the receiver
— plug the computer into a different outlet so that computer and receiver are on different branch circuits.

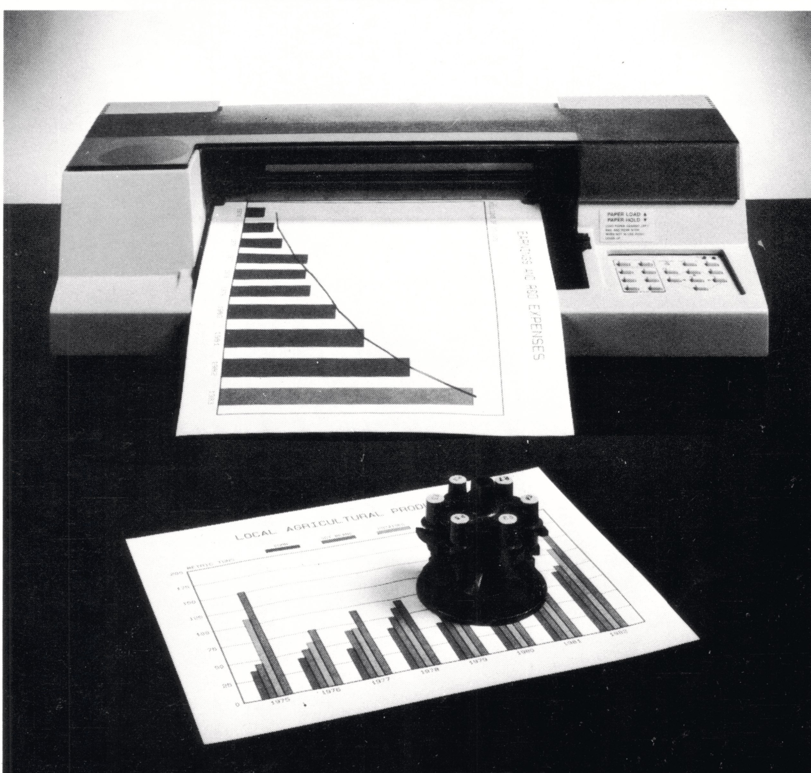
If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

'How to Identify and Resolve Radio-TV Interference Problems'.

This booklet is available from the US Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4."

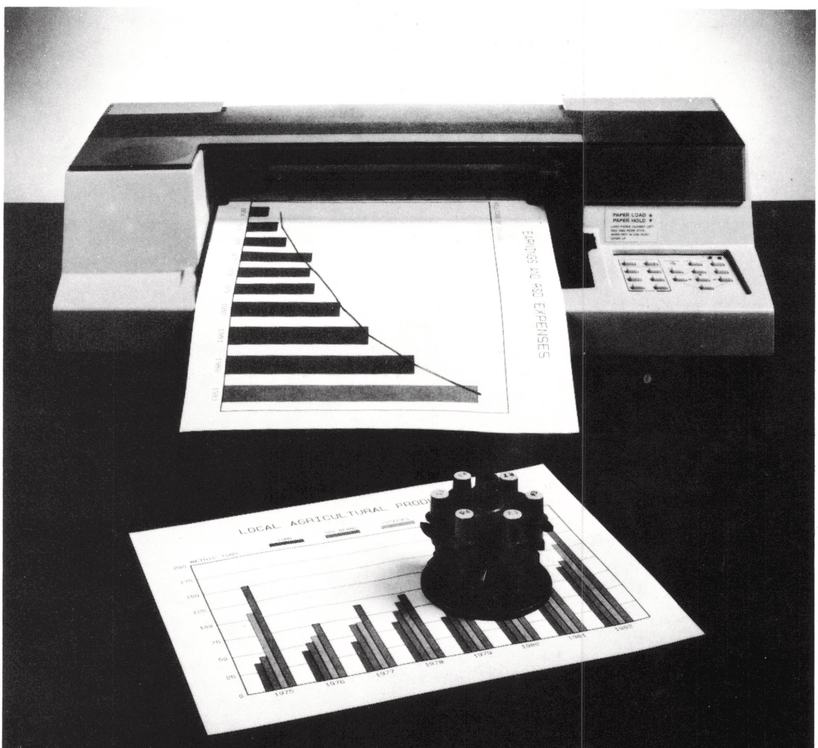
INTERFACING AND PROGRAMMING MANUAL

HP 7475A
Graphics Plotter



INTERFACING AND PROGRAMMING MANUAL

HP 7475A
Graphics Plotter



16399 W. Bernardo Drive, San Diego, CA 92127-1899

Manual Summary

Chapter 1: Getting Started

Contains information concerning manual usage, a description of the plotter, its interfaces, the HP-GL language, and four instructions.

Chapter 2: Establishing Boundaries and Units

Explains the concept of plotting area, plotter and user units, scaling, and the instructions used to set and output the scaling points and window, to scale the plotting area, and to rotate the coordinate system.

Chapter 3: Controlling the Pen and Plotting

Describes the instructions for pen control, vector plotting, and for defining and filling rectangles and arc segments.

Chapter 4: Enhancing the Plot

Describes instructions for drawing tick marks and differentiating traces.

Chapter 5: Labeling

Describes the instructions used in labeling to set direction, size, and slant of characters, as well as instructions for character set and label terminator selection and for designing your own characters.

Chapter 6: Digitizing

Describes the instructions used to digitize with the plotter and demonstrates how to check for the presence of a digitized point.

Chapter 7: Obtaining Information from the Plotter

Describes the instructions used to obtain information about pen position, errors, and capabilities of the plotter.

Chapter 8: Putting the Commands to Work

Three examples illustrating the procedures to be followed to draw labels and plot data using HP-GL instructions.

Chapter 9: HP-IB Interfacing

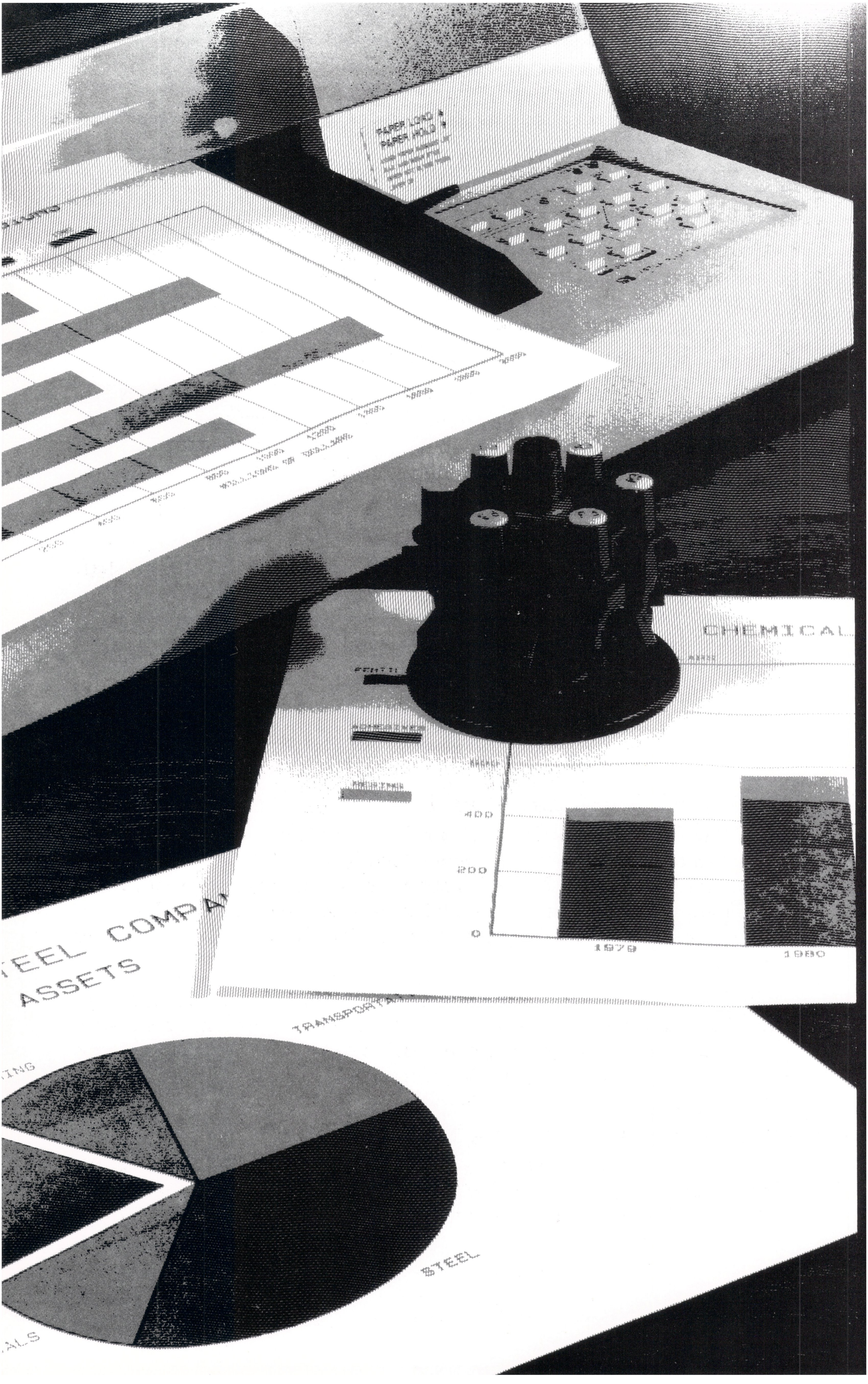
Summarizes operation of the plotter with the Hewlett-Packard Interface Bus (HP-IB) and explains the methods for addressing and sending and receiving data over the interface bus.

Chapter 10: RS-232-C/CCITT V.24 Interfacing

Describes how to connect the plotter with a terminal and/or computer, summarizes the methods for establishing a handshake protocol between the plotter and computer, and explains the device control instructions that are used to set up and control the handshake protocol.

Appendix A: An HP-IB Overview

Provides an overview of the Hewlett-Packard Interface Bus (HP-IB).


PAPER LOAD
PAPER HOLD
CHEMICAL
400
200
0
1979
1980
ASSETS
STEEL

Chapter 1
Getting Started

What You'll Learn in This Chapter

This chapter explains how to use this manual and other manuals you may need or find useful. In addition, this chapter describes:

- The 7475 Graphics Plotter's features
- Its two interfaces
- The plotter's language and syntax
- Four instructions from the plotter's language, HP-GL (Hewlett-Packard Graphics Language).

HP-GL Instructions Covered

DF The Default Instruction
IN The Initialize Instruction
IM The Input Mask Instruction
PS The Paper Size Instruction

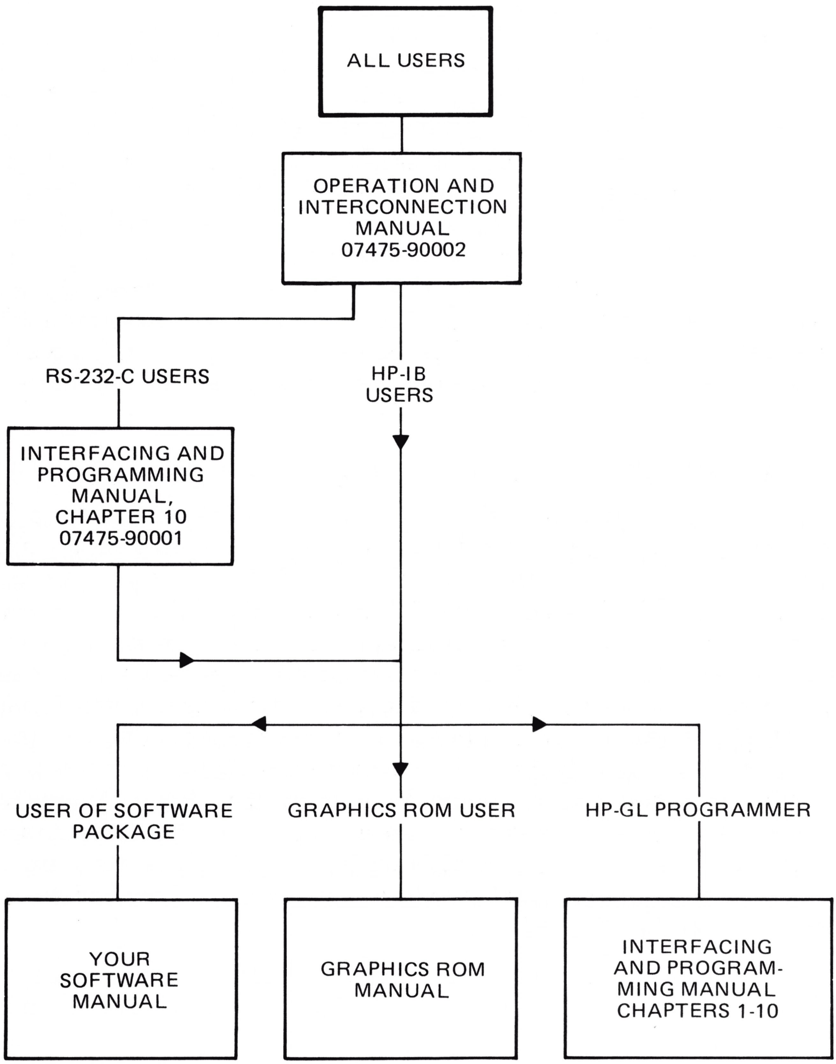
Terms You Should Understand

HP-GL — Hewlett-Packard Graphics Language — the two-letter-mnemonic graphics language understood by the 7475 plotter and other HP graphics devices. The instruction's mnemonic is suggestive of its role. For example, PA is used to plot to absolute coordinates, SP is used to select a pen, and DR is used to establish the relative direction of labeling.

HP-IB — Hewlett-Packard Interface Bus — HP's implementation of the IEEE standard 488-1978 digital interface for programmable instrumentation is commonly found on HP desktop computers and some larger computers. The HP-IB interface is standard on the Option 002 plotter.

RS-232-C/CCITT V.24 Interface — another popular standardized interface. It is commonly found on large computers, personal computers, and where communication between a terminal and a computer over telephone lines is required. This interface is standard on the Option 001 plotter.

Follow the documentation road map below:



A Brief Look at the 7475 Plotter

The HP 7475 Graphics Plotter is a vector plotter which produces high quality, multicolor graphics plots on four sizes of drawing media:

ISO A4 (210 × 297 mm)	ANSI A (8½ × 11 in.)
ISO A3 (297 × 420 mm)	ANSI B (11 × 17 in.)

Plotter Instruction Set (Continued)

Instruction		Description
SM	c [c]	Symbol mode
SP	n [i]	Select pen
SR	width [d], height [d]	Relative character size
SS		Select standard character set
TL	tp [d] (,tn [d])	Tick length
UC	(pen [i],) X [d], Y [d], pen [i] (, . . .)	User defined character
VS	v [d]	Select velocity v
WG	radius [i/sd], start angle [i], sweep angle [i] (,chord angle [i])	Shade wedge
XT		X-axis tick
YT		Y-axis tick

[c] = character format

[i] = integer format, −32 768 to +32 767

[sd] = scaled decimal format, −32 768.0000 to +32 767.9999

How to Use the Examples in This Manual

The examples in this manual are designed primarily to show the use of the instruction with which they appear. If you are new to programming, try entering and running some examples on your computer. You might then wish to change some parameters in an instruction and rerun the plot. The examples are presented in two ways, either as complete programs or as listings of only the pertinent HP-GL strings.

Examples Presented as Complete Programs

Some examples are presented as complete programs, written in a version of Microsoft® GW BASIC for MS™-DOS operating systems. This BASIC is used by several popular personal computers.

Be sure you have established the proper handshaking protocol before running these programs. The 7475A Operation and Interconnection Manual contains configuration examples for several personal computers. If your computer is not listed there, refer to your computer's documentation. It will tell you how to establish communication between your computer and the plotter.

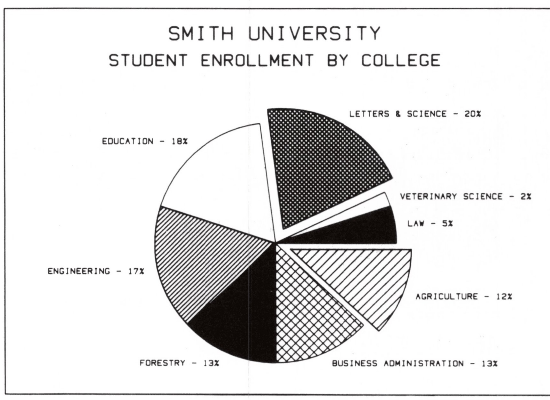
Following is a simple example of the way complete program appears in this manual. You will always need a configuration statement at the

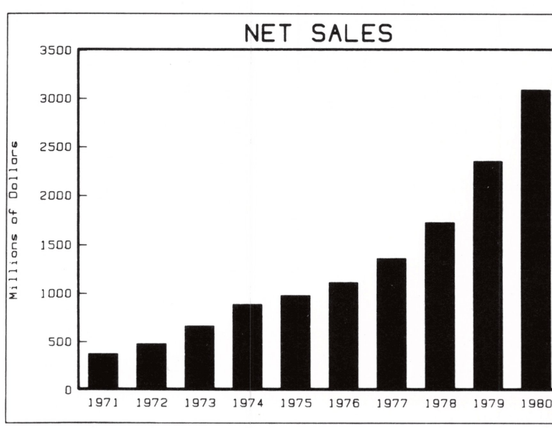
Looking Ahead

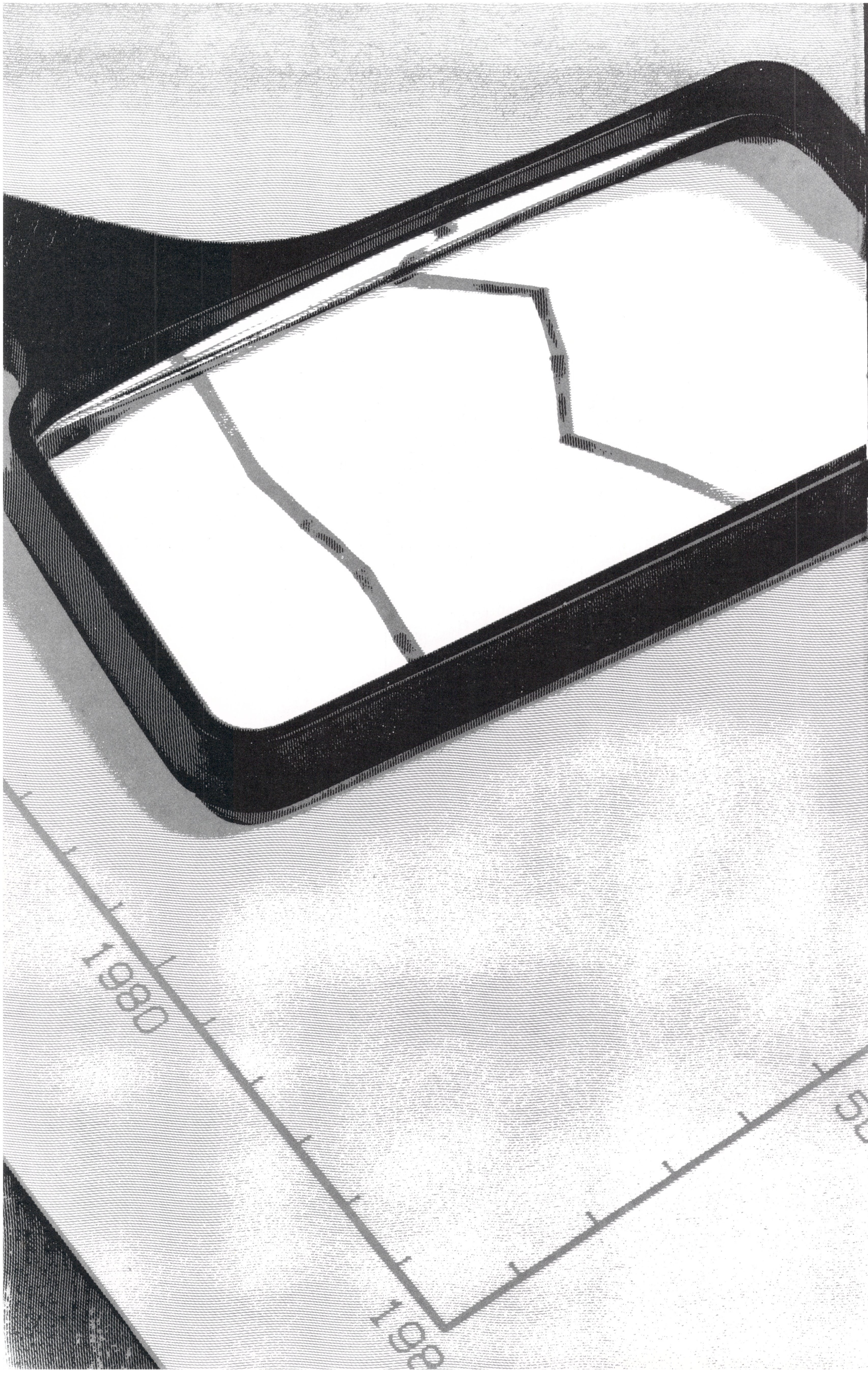
Of course you want to use your plotter to create high quality graphic plots. Most data display plots fall into one of three broad classes: line graphs, bar graphs, or pie charts. Chapter 8 contains sample programs for a line graph, a bar graph, and a pie chart.

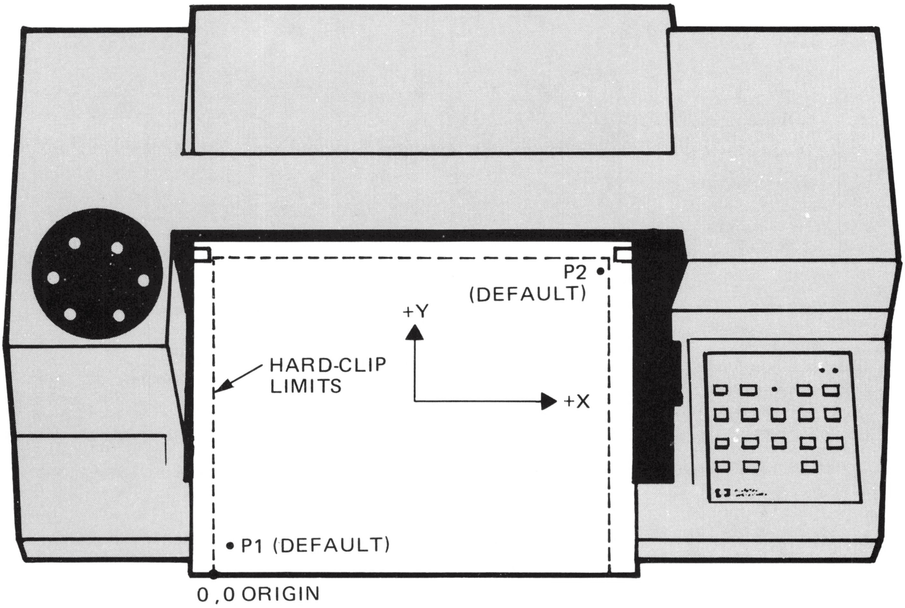
Pie charts are an effective way to show parts of a whole entity; the slices of the pie are the component parts. The pie chart shown here has some segments "exploded" for emphasis. To construct a pie chart, the data is computed as a percentage of the total and each data value is converted to the appropriate segment of a full 360-degree circle. To create a simple pie chart, you can use the WG and EW instructions to draw and fill segments of a circle (arcs) as shown in Chapter 8. Additional information on drawing circles is available under the CI instruction, and on shading and edging the segments of pie charts, under the WG and EW instructions in Chapter 3.

There are three types of bar graphs: simple bar graphs, stacked bar graphs, and clustered bar graphs. The simple bar graph here shows that sales are increasing. Bar graphs are essentially a collection of rectangles. Each of these rectangles is filled; refer to the FT, RA, and RR instructions in Chapter 3 to learn how to create a filled or hatched rectangle. A stacked bar might be used to show these same sales data broken down into sales by region. Portions of each bar would be colored or shaded differently to show the sales in each region. A sample stacked bar program is shown in Chapter 8. Another way of showing sales by region would be to use a separate bar for each region and to "cluster" all the bars for one year together with a larger space between each cluster of bars.

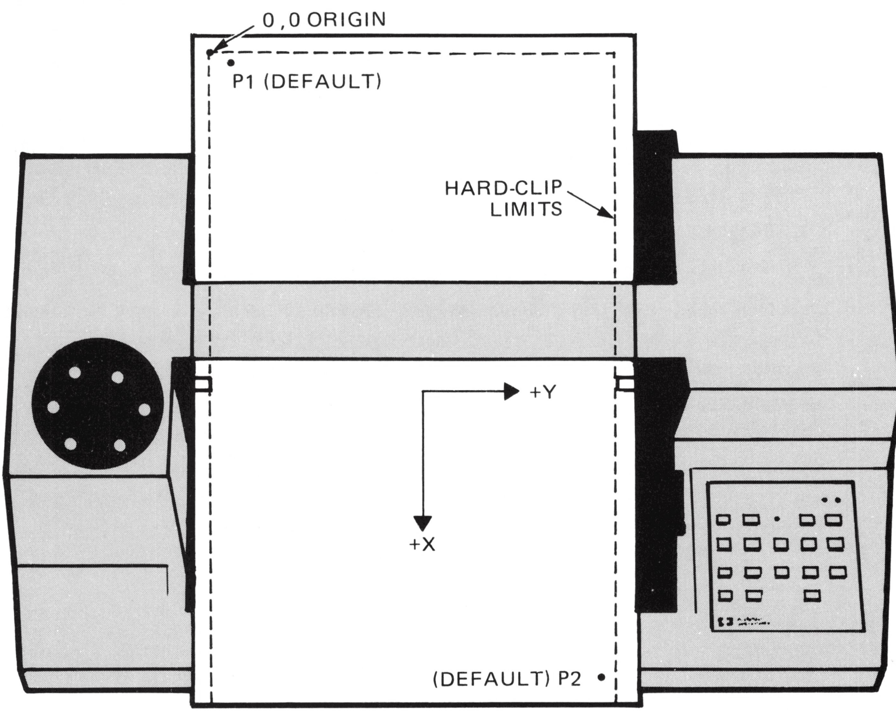




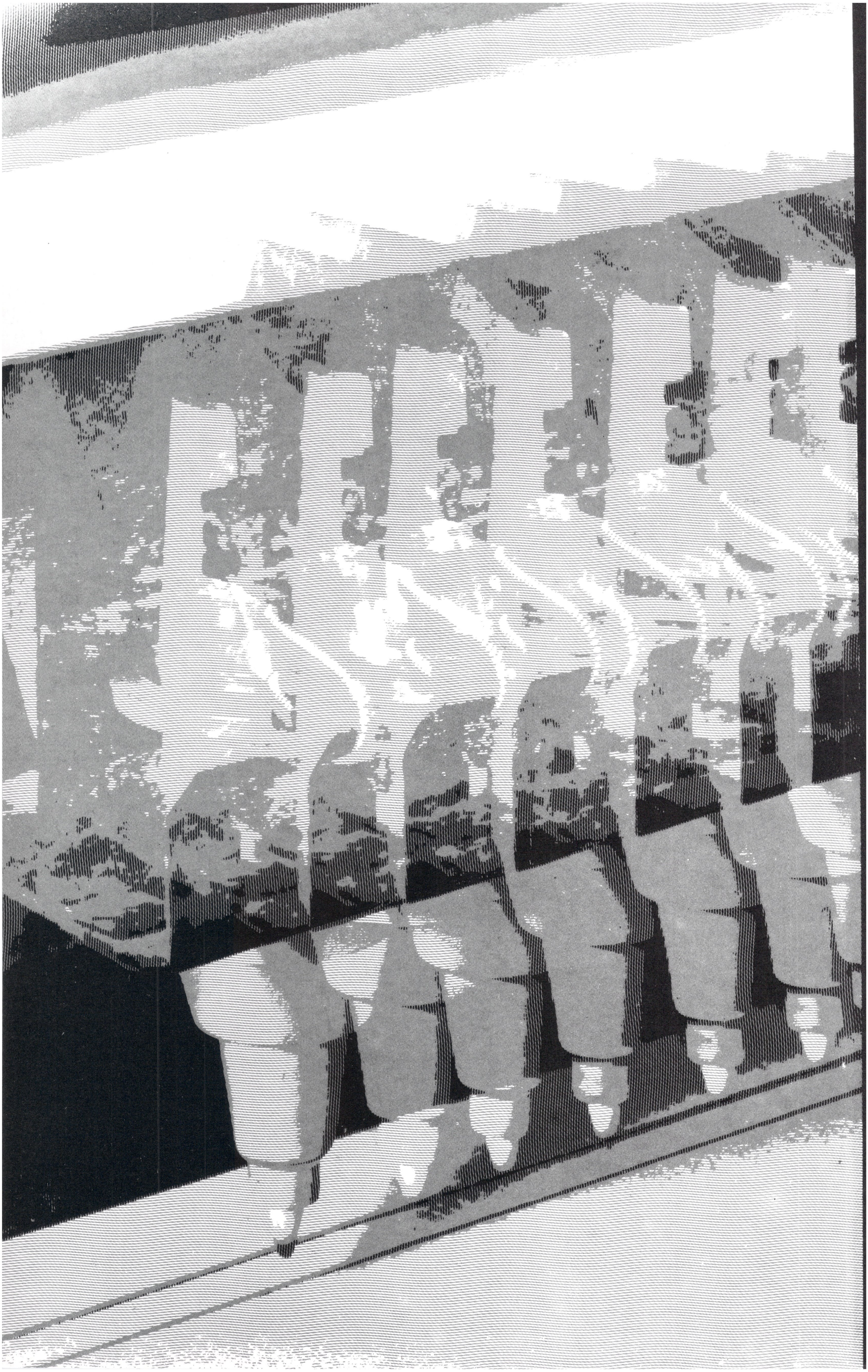

1980



Default Orientation of Plotter Coordinate System (A/A4 Paper)



Default Orientation of Plotter Coordinate System (B/A3 Paper)



Chapter **3**

Controlling the Pen and Plotting

What You'll Learn in This Chapter

Now that you understand the unit systems in which data can be represented, you are ready to create plots. In this chapter, you will learn how to select or change pens, how to set and change pen velocity, how to raise and lower the pen, and how to plot. You will learn how to plot to absolute X,Y coordinates or to plot relative to the last pen position. You will also learn how to send variables as parameters of plot instructions; this will enable you to write general purpose graphics programs. Finally, you will learn how to define and fill rectangles and arc segments.

HP-GL Instructions Covered

SP	The Select Pen Instruction
VS	The Velocity Select Instruction
PU/PD	The Pen Up/Down Instructions
PA	The Plot Absolute Instruction
PR	The Plot Relative Instruction
CI	The Circle Instruction
AA	The Arc Absolute Instruction
AR	The Arc Relative Instruction
FT	The Fill Type Instruction
PT	The Pen Thickness Instruction
RA	The Shade Rectangle Absolute Instruction
EA	The Edge Rectangle Absolute Instruction
RR	The Shade Rectangle Relative Instruction
ER	The Edge Rectangle Relative Instruction
WG	The Shade Wedge Instruction
EW	The Edge Wedge Instruction

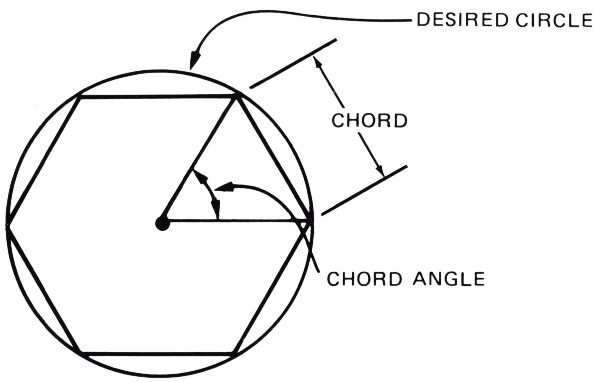
Terms You Should Understand

Absolute Plotting — plotting to a point whose location is specified relative to the origin (0,0). When the PA instruction is used to plot to a point, the pen always moves to the same point on the plotting surface, no matter where the pen was before the move.

Relative Plotting — plotting to a point whose location is specified relative to the current pen position. The point moved to then becomes

point; a circle with a negative radius starts at the 180-degree point. The current pen position is the center of the circle. If scaling is off, the radius is in plotter units. If scaling is on, the radius is in user units. If user units are not the same size in the X- and Y-directions, ellipses will be drawn.

The chord angle parameter is in integer format and governs the smoothness of the circle. It is interpreted as degrees and sets the maximum angle subtended by a chord that is drawn to represent an arc segment of the circle, as shown below. The actual angle used may be changed by the plotter so that all chords are the same length. The sign of the parameter is ignored, except to set the maximum in-range limit to −32 768 or +32 767.

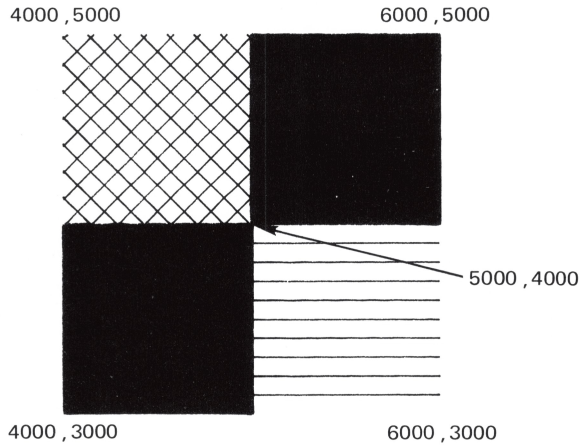


The most useful chord angle values range from 0 to 180; where 0 produces the smoothest circle and larger numbers progressively reduce the number of chords used. Values from 180 to 360 work just the opposite; i.e., larger numbers progressively increase the number of chords used and 360 produces the smoothest circle. This pattern follows modulo 360 through the permitted range of −32 768 to +32 767. Specifying out-of-range parameters sets error 3 and the instruction is ignored.

The following strings of HP-GL instructions, when sent to the plotter using your computer's output statments, show the effect of different chord angles.

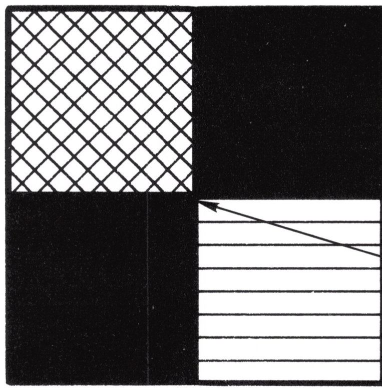
```
"IN;SP1;IP3650,2325,6650,5325;"
"SC-100,100,-100,100;"
"PA-50,40;CI30,45;"
"PA50,40;CI30,30;"
"PA-50,-40;CI30,15;"
"PA50,-40;CI30,5;"
"SP0;"
```

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20 PRINT #1, "IN;SP1;PA5000,4000;"
30 PRINT #1, "PT.3;FT1;RA4000,3000;"
40 PRINT #1, "FT3,100;RA6000,3000;"
50 PRINT #1, "FT2;RA6000,5000;"
60 PRINT #1, "FT4,100,45;RA4000,5000;"
70 PRINT #1, "SP0;"
80 END
```



10	configuration statement; change this statement as necessary for your computer.
20	initializes the plotter, selects a pen (pen 1), and sets the starting position.
30	selects pen thickness, fill type 1 (solid fill, bidirectional), and sets the X,Y coordinates for the first rectangle.
40	selects the fill type and spacing, and sets the X,Y coordinates for rectangle 2. Notice that you do not need to repeat the pen thickness instruction since it will remain in effect until you select a new pen or a new pen thickness.
50	selects the fill type and sets the X,Y coordinates for rectangle 3.
60	selects a new fill type, spacing, and angle, and sets the X,Y coordinates for rectangle 4.
70	puts the pen back in the carousel.

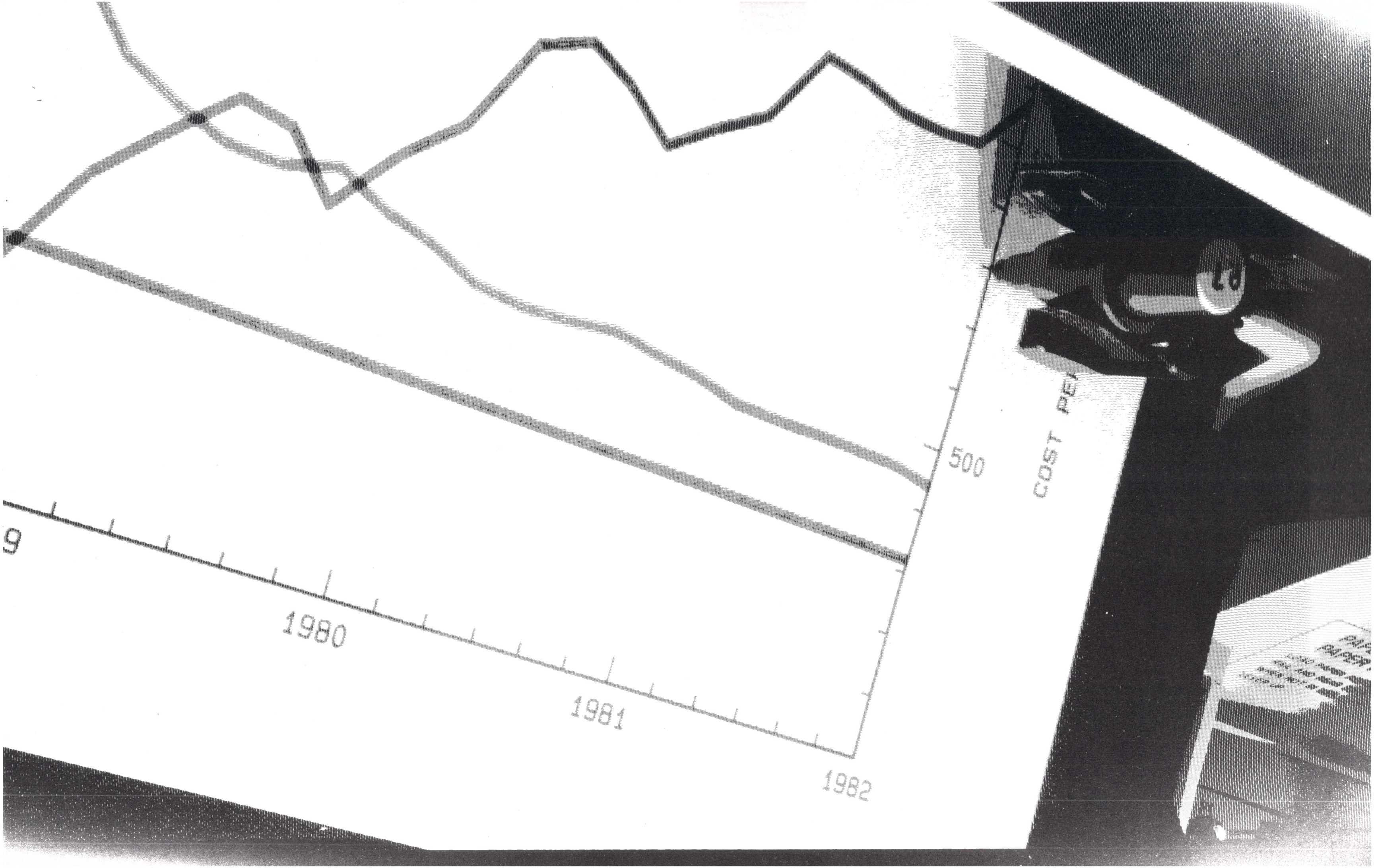
```
80 PRINT #1,  "SP3;EA6000,5000;"
90 PRINT #1,  "SP6;FT4,100,45;RA4000,5000;"
100 PRINT #1, "SP3;EA4000,5000;"
110 PRINT #1, "SP0;"
120 END
```



10	configuration statement; change this statement as necessary for your computer.
20	initializes the plotter, selects a pen (pen 1), and sets the starting position.
30	selects pen thickness, fill type 1 (solid fill, bidirectional), and sets the X,Y coordinates for the first rectangle.
40	selects a new pen (pen 3) and edges the first rectangle.
50	selects a new pen, a new fill type, and sets the X,Y coordinates for rectangle 2.
60	selects a new pen and edges rectangle 2.
70	selects a new pen, new fill type, and sets the X,Y coordinates for rectangle 3.
80	selects a new pen and edges rectangle 3.
90	selects a new pen, new fill type, spacing and angle, and sets the X,Y coordinates for rectangle 4.
100	selects a new pen and edges rectangle 4.
110	puts the pen back in the carousel.

The Shade Rectangle Relative Instruction, RR

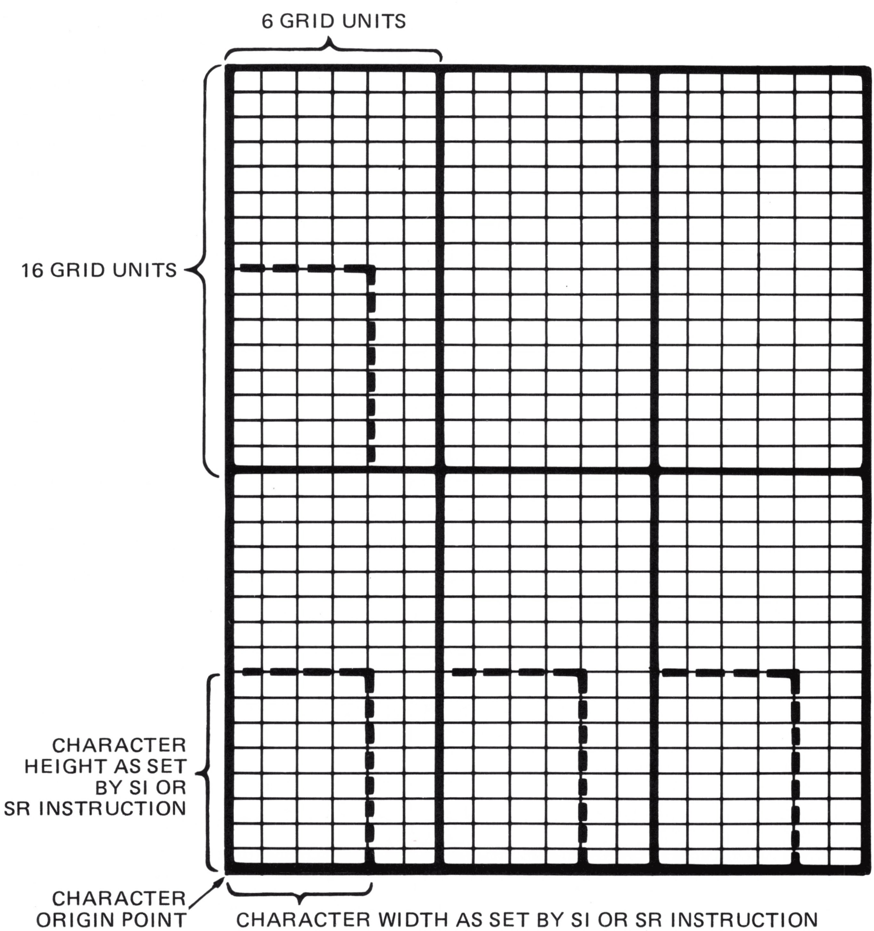
DESCRIPTION The shade rectangle relative instruction, RR, can be used to define and shade a rectangle using relative coordinates.


COST PE
500
1980
1981
1982


XY
50
1980
79
JAN

SYNTAX *UC* (pen control,)X-increment,Y-increment,(pen control,)
(X-increment,Y-increment,) . . . , . . . , terminator
or
UC terminator

EXPLANATION Each segment of the character is drawn on a character grid. This grid is established on each character-space field by dividing it into 6 horizontal units and 16 vertical units. The size of the character-space field and, hence, the grid unit is set by the current size instruction. The size of the character-size space field and, thus, the grid is always twice the current character height and 1½ times the current character width. To draw a user-defined character the same size as a character drawn with a label instruction, design the user-defined character in the lower-left corner of the grid with a width of four grid units and a height of eight grid units.



Character Grid

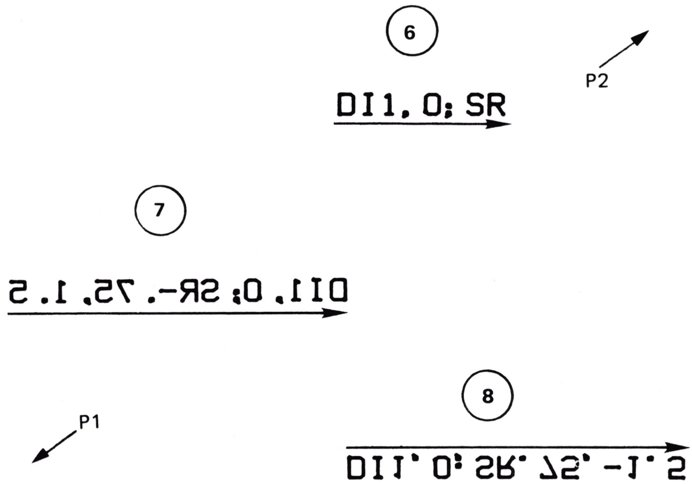
The following example generates a Σ symbol which is the same size as an uppercase letter. For comparison, an "E" is drawn with the label instruction. The example shows how size instructions affect both user-defined characters and labeled characters. The HP-GL instructions appear in quotation marks in the BASIC PRINT statements. Other BASIC statements, FOR and NEXT, are included in this example.

```
20  PRINT #1, "IN;SP1;PA1000,1000;"
30  FOR A=.19 TO .89 STEP .1
40  PRINT #1, "SI",A,A*1.4
50  PRINT #1, "UC4,7,99,0,1,-4,0,2,-4,-2,-4,4,0,0,1;"
60  NEXT A
70  PRINT #1, "PA1000,1750;"
80  FOR B=.19 TO .89 STEP .1
90  PRINT #1, "SI",B,B*1.4
100 PRINT #1, "LBE"+CHR$(3)
110 NEXT B
```

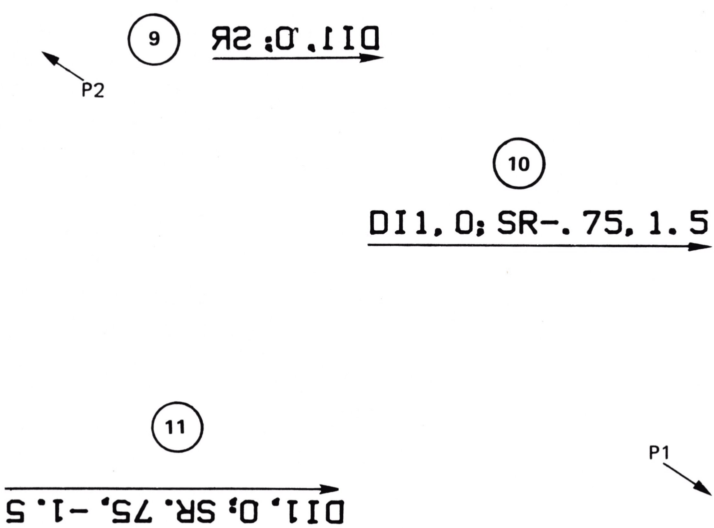
User-defined characters need not fit into a single character-space field. In the next example, the user-defined character takes up more than one character space. Since this character is to be followed by a label, a CP instruction must be added to move the current pen position beyond the limits of the user-defined character. The reference point for parameters of CP instructions is the pen position at the completion of the user defined character, one character-space field to the right of the origin of the user-defined character.

```
"IN;SP1;PA1000,5000;SI.25,.4;"
"UC0,4,99,1.75,0,1.5,4,3,-8,3,8,3,-8,3,8,
 3,-8,1.5,4,1.75,0;"
"CP3.25,0;LB1000 ohms"+CHR$(3)
```

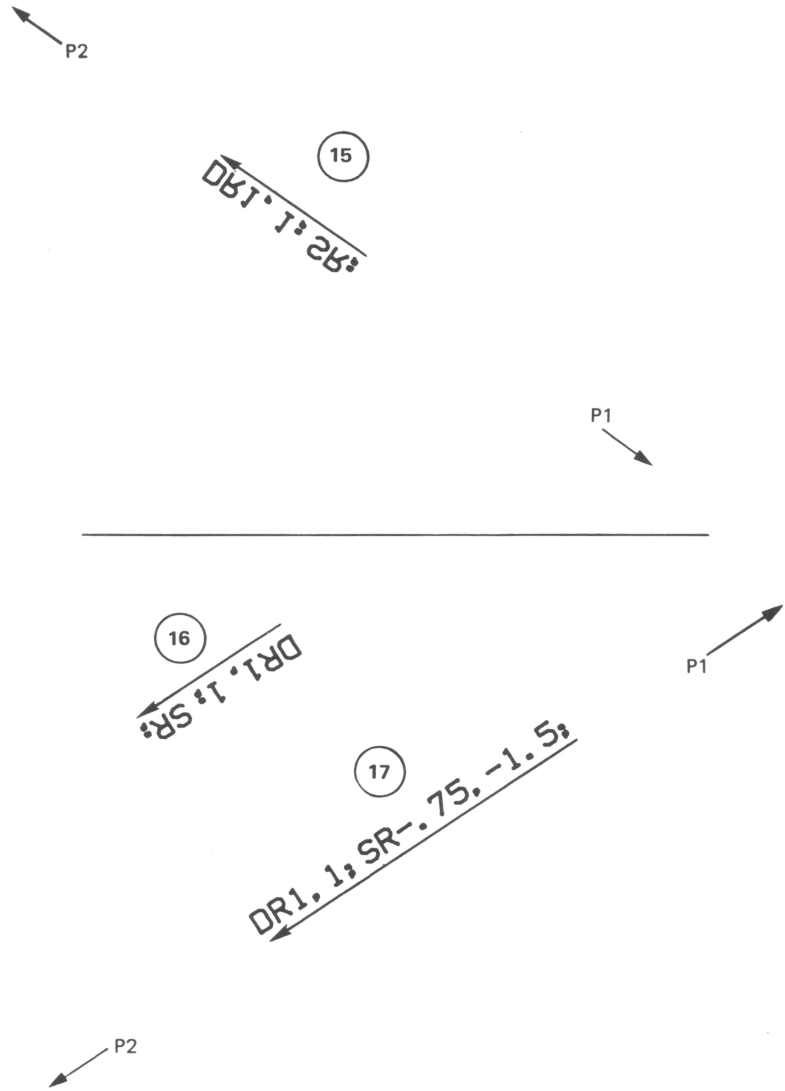
User-defined characters are drawn using the current character size, slant, and direction. It is also possible to change the size of a user-defined character by changing each X- or Y-increment parameter by a



In the next three illustrations, P1 and P2 have been changed so P1 is lower right and P2 is upper left. Hence ($P2_x - P1_x$) is negative and anything with a positive SR width parameter is mirrored right-to-left, e.g., illustrations 9 and 11. The effect of the negative width parameter in illustration 10 is cancelled by the negative difference ($P2_x - P1_x$).



In the next illustrations, P1 and P2 have both been flipped so P1 is upper right and P2 is lower left. Now any positive parameter causes mirroring and any negative parameter cancels mirroring. This can be seen in examples 12, 13, and 14.


P2
15
DR1,1;SR;
P1
16
DR1,1;SR;
P1
17
DR1,1;SR-.75,-1.5;
P2

Advanced Programming Tips

When drawing labels, you often wish to position them precisely in relation to a specific point. Unless positioned differently by the programmer, labels are written beginning at the current pen position which marks the baseline of the label.

The following BASIC program illustrates various ways to center labels. The program uses the BASIC function LEN to find the length of the string. This length is used to determine horizontal adjustments, i.e., how many character-space widths the pen must be moved to achieve the desired positioning. Vertical moves are in terms of character-space heights. Since an uppercase letter is half the height of a character space, a vertical movement of one-quarter character space down will center uppercase letters on the point; notice the parameter is negative. A parameter of –0.5 will cause the top of uppercase letters to be level with the point.

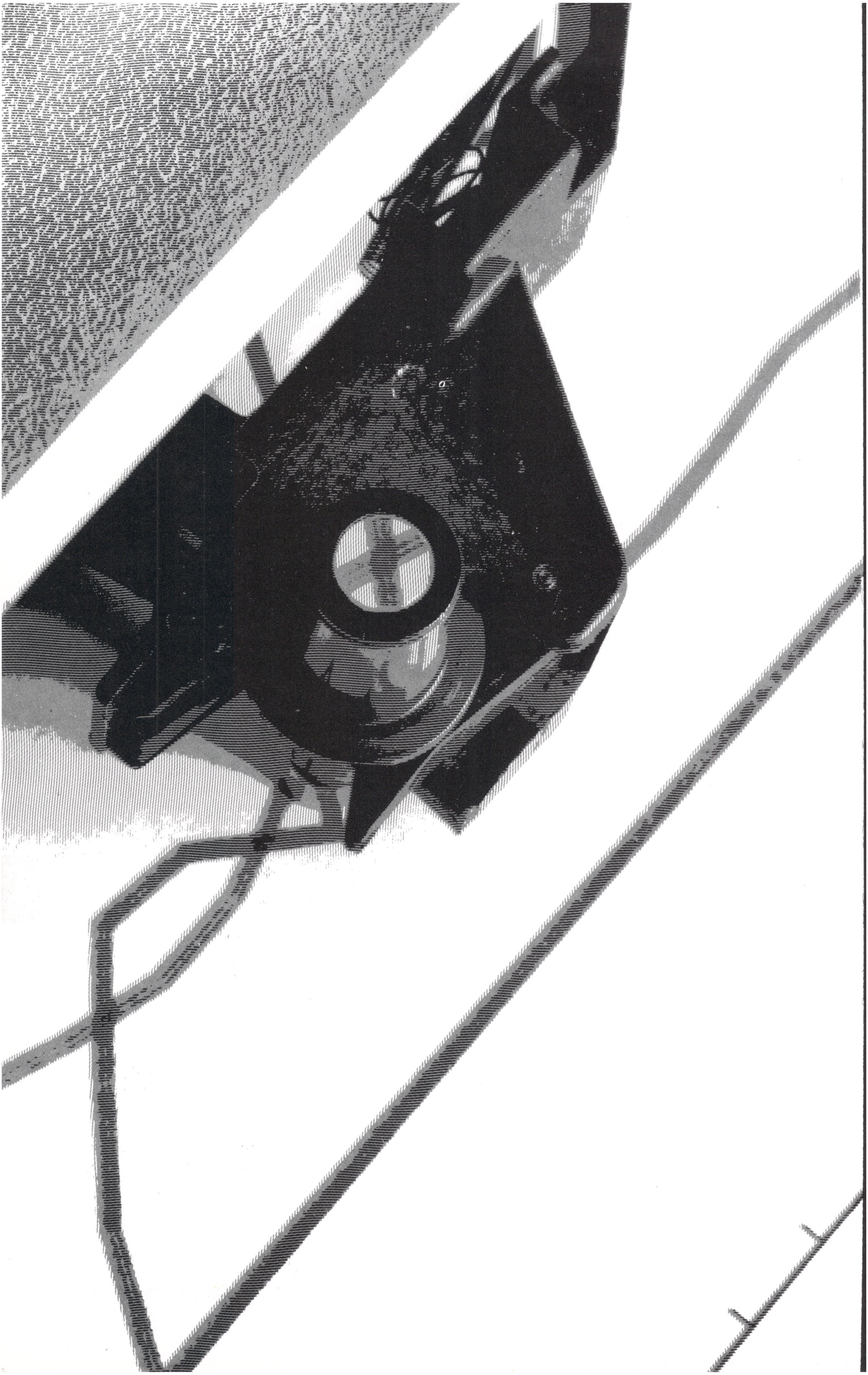
Symbol mode plotting, with an * as the symbol, has been used here to show pen position at the start of the label instruction. The character plot instruction which positions the label is shown above each label.

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20 DIM A$(40),B$(40),C$(40)
30 A$ = "THIS LABEL IS RIGHT JUSTIFIED"
40 PRINT #1, "IN;SP1;SM*;PA6000,5500;"
50 PRINT #1, "CP";-LEN(A$);"0;LB";A$+CHR$(3)
60 B$ = "THIS LABEL IS CENTERED BELOW THE POINT"
70 PRINT #1, "PA4500,5000;"
80 PRINT #1, "CP";-LEN(B$)/2;"-.5;LB";B$+CHR$(3)
90 C$ = "VERITCALLY CENTERED LABEL"
100 PRINT #1, "PA2750,4500;"
110 PRINT #1, "CP0,-.25;LB";C$+CHR$(3)
120 END
```

```
"CP";-LEN(A$);"0;"
        THIS LABEL IS RIGHT JUSTIFIED*

"CP";-LEN(B$)/2;"-.5;"
   THIS LABEL IS CENTERED BELOW THE POINT

"CP0,-.25;"
          *VERTICALLY CENTERED LABEL
```



Chapter 6
Digitizing

What You'll Learn in This Chapter

The plotter can be used as a digitizer as well as a plotter. Digitizing consists of moving the pen or digitizing sight to a point on the plotting surface, entering the point, and sending the coordinates of that point to the computer. This chapter describes the three instructions used in digitizing, and contains a discussion of the steps required by a computer program for digitizing; sample programs are also included. Included in the discussion are three different methods of assuring that a point has been entered. The method you will use will depend on your application and your interface (HP-IB or RS-232-C).

HP-GL Instructions Covered

DP The Digitize Point Instruction
DC The Digitize Clear Instruction
OD The Output Digitized Point and Pen Status Instruction

Terms You Should Understand

Digitizing — converting information, in this case pen position and up/down status, to digital information so that it can be understood by the computer.

Output Terminator — the character or characters sent by the plotter at the end of the response to an output instruction. It is interface-dependent.

Preparing Your Plotter for Use as a Digitizer

A plotter with an HP-IB interface must be set to an address less than 31 because the plotter cannot send the coordinates of a digitized point to the computer when it is in listen-only mode.

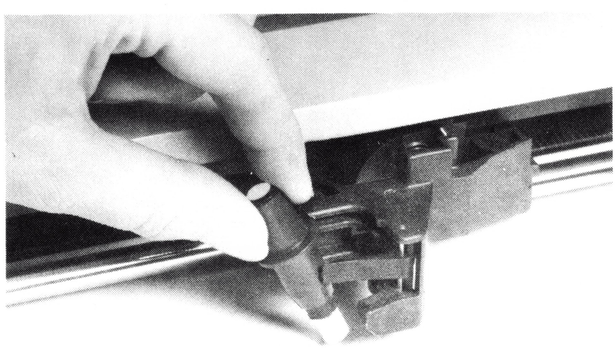
Use of a digitizing sight, available as an accessory with the 7475, is recommended. The sight should be loaded manually into the pen holder itself. Slip the digitizing sight gently into the pen holder just as you would slip in a pen.

CAUTION

The sight should not be stored in a pen stall; do not store using front panel buttons or an SP command. Remove the sight from the pen holder before raising the PAPER LOAD lever since the sight would be stored automatically when the lever is raised.

To remove the sight from the pen holder, slip the sight out of the pen holder.

The sight is used in the pen down position.



Loading the Sight

The Digitize Point Instruction, DP

DESCRIPTION The digitize point instruction, DP, provides the means to digitize points on the plotter.

USES This instruction can be used to input data for a graphics program or obtain the coordinates of a point or points on the plot.

SYNTAX *DP* terminator

EXPLANATION No parameters are used.

When the DP instruction is received, automatic pen lift is suppressed, the current front-panel paper-size light blinks, and the plotter is ready to have a digitized point entered by pressing **ENTER** on the front panel.

When **ENTER** is pressed, the X- and Y-coordinates of that point and pen up/down status are stored for retrieval by the OD instruction. Pressing **ENTER** sets bit position 2 of the status byte, indicating a digitized point is available for output.

After **ENTER** has been pressed, automatic pen lift is reactivated, and the paper-size light stops blinking.

The Digitize Clear Instruction, DC

DESCRIPTION The digitize clear instruction, DC, provides a means to terminate digitize mode.

USES This instruction can be used to terminate digitize mode without entering a point. If you are using an interrupt routine in a digitizing program to branch to some other plotting function, you could use DC to clear digitize mode immediately after branching.

SYNTAX *DC* terminator

EXPLANATION No parameters are used.

When the DC instruction is received, digitize mode is terminated, and the paper-size light stops blinking. Automatic pen lift is reactivated.

The Output Digitized Point and Pen Status Instruction, OD

DESCRIPTION The output digitized point and pen status instruction, OD, is used to output the X- and Y-coordinates and pen up/down status associated with the last digitized point.

USES This instruction is used after DP and **ENTER** in all digitizing applications to return the coordinates of the digitized point to the computer.

SYNTAX *OD* terminator

EXPLANATION No parameters are used.

The timing of output depends on the plotter's interface (HP-IB or RS-232-C). Refer to A Brief Word about Plotter Output in Chapter 7 for more information.

The pen position and status are output to the computer as integers in ASCII in the form:

X,Y,P TERM

where X is the X-coordinate of the digitized point in plotter units,
Y is the Y-coordinate of the digitized point in plotter units,
P is the pen status when the point was entered (0 = pen up, 1 = pen down), and
TERM is the output terminator for your system (refer to Chapter 7).

The ranges of the X- and Y-coordinates are the hard-clip limits of the plotter as determined by the setting of the paper switches.

Upon receipt of the OD instruction by the plotter, bit position 2 of the output status byte is cleared.

Digitizing with the 7475

When using the plotter as a digitizer, it is important to ascertain that a point has been entered before an attempt is made to retrieve that point using the OD instruction. There are three methods for doing this.

Manual Method

The first method, which might be called the manual method, is easiest to understand. It is not efficient in applications where many points will be entered, or in an RS-232-C environment where the mainframe is not adjacent to the plotter or where human intervention in program execution is not possible. The steps in this method are as follows:

1. In a program, send a DP instruction to the plotter. Follow the DP instruction immediately with a statement that will cause the program to display or print a message prompting you to enter a point. Follow the prompt with a statement that will cause the program to pause until instructed to continue. The BASIC statement PAUSE will accomplish this.
2. Move the digitizing sight (pen) to the point to be entered, using front-panel buttons. Final positioning should be done with the sight (pen) down.
3. Press **ENTER** on the plotter's front panel. Now resume running of the program. This is done on HP desktop computers by pressing the key marked **CONTINUE** or **CONT**.

4. The program step following the pause will now be executed. The next steps of the program, in order, should be an OD instruction to the plotter, a read statement by the computer to read the X- and Y-coordinates and the pen status, a statement to remove the prompt (requesting you to enter a point) from the screen, and then steps to process the digitized data in the appropriate manner.

Using this method, there is no need to monitor the status byte because the program does not proceed to the OD instruction until the user enters a point and causes the program to resume.

A simpler procedure, using OA or OC instead of OD, can also be used. It omits the DP in step 1 and pressing **ENTER** in step 3. Using the shorter procedure with OC makes it possible to obtain coordinate values in user units. Refer to Chapter 7.

A short program to digitize a single point and display the coordinates and pen status is given below.

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20 PRINT #1, "DP;"
30 PRINT "Enter a point, then press RETURN"
40 INPUT  N$
50 PRINT #1, "OD;"
60 INPUT #1, X,Y,P
70 PRINT X, Y, P
80 END
```

Monitoring the Status Byte

The second method monitors bit position 2 (the third least significant bit) of the plotter's status byte, which is set when a digitized point is available. Refer to the Output Status Instruction, OS, in Chapter 7 for more information.

There are a variety of ways to monitor bit position 2, depending on the instructions available in the computer you are using. The status byte can be operated on arithmetically to check for the availability of a digitized point. Executing successive divisions of a number by a power of two and checking the answer for an odd or even integer is a common way of monitoring bits without converting the number to binary form. The following example uses this method.

Example — Digitizing by Monitoring the Status Byte

The following sequence of BASIC instructions will check the proper bit of the status byte. In line 50, the INPUT# statement reads the status byte into a variable called Status. (INT is a function that returns the integer portion of a number.)

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20 PRINT #1, "DP;"
30 PRINT "Enter a point by pressing ENTER"
40 PRINT #1, "OS;"
50 INPUT #1, STATUS
60 STATUS = INT(STATUS/4)
70 IF STATUS = INT(STATUS/2)*2 THEN 40
80 PRINT #1, "OD;"
90 INPUT #1, X,Y,P
100 PRINT X,Y,P
110 END
```

Program Explanation

10	configuration statement
20	prepares plotter to accept a digitized point
30	prompts you to enter a point on the plotter and press **ENTER** on the plotter.
40	sends the output status instruction
50	reads the status
60	shifts bits right by two positions
70	if a point hasn't been obtained, reads status again
80	outputs the digitized point
90	reads X, Y coordinates and pen status (up/down)
100	displays X, Y coordinates and pen status

Example — Digitizing Many Points

In many applications, a large number of points need to be digitized. When the computer is used to monitor bit position 2, the data points may or may not be processed immediately. Generally, you need to allocate space for the toal number of points to be digitized. Then, you can establish a loop to process the total number of points, calling a subroutine each time to check that a point has been entered.

A complete BASIC program follows. When prompted to enter a point, use the cursor keys to move the digitizing sight to the desired position. Now press the **ENTER** button on the plotter. Continue for all 25 points. Their coordinates will be displayed on the computer's screen after they have all been entered.

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20 DIM X(25),Y(25),P(25)
30 FOR C = 1 TO 25
40    PRINT #1,"DP;"
50    PRINT "ENTER POINT ";C
60    GOSUB 140
70    PRINT #1,"OD;"
80    INPUT #1,X(C),Y(C),P(C)
90 NEXT C
100 FOR C = 1 TO 25
110    PRINT X(C),Y(C),P(C)
120 NEXT C
130 END
140 REM Check bit 2 for available digitized point
150 PRINT #1,"OS;"
160 INPUT #1,STATUS
170 STATUS = INT(STATUS/4)
180 IF STATUS = INT(STATUS/2)*2 THEN 150
190 RETURN
```

HP-IB Interrupts and Polling

A third method can be used by advanced programmers thoroughly familiar with the HP-IB interface, polling techniques, and interrupts. It should only be used when the computer can perform useful tasks while waiting for the digitized point to be entered. This method involves setting a value of 4 in the S-mask of the IM instruction, e.g., IM 223,4,0; to cause the plotter to generate an RQS (service request) when a digitized point is available. With an interrupt routine enabled for service requests, the computer can send a DP instruction to initiate digitizing, and then proceed with some other task until the digitized point is entered. When the point is available, the computer is interrupted by the RQS, and program execution branches to the routine to process the digitized data. This routine could simply send an OD instruction and read the digitized point, or it could perform bit checking of the plotter status byte if multiple S-mask values have been specified to generate the RQS. The status byte can be obtained by serial polling or simply by sending an OS instruction. Because interrupts and polling are highly machine-dependent and beyond the scope of this manual, no examples are given.

0 PRINTER IS 705
0 PRINT "OP;"
0 ENTER 705 ;
0 PRINT "OE;"
0 ENTER 705 ;
0 PRINT "OI;"
0 ENTER 705
0 DISP A;B;C
0 END 279
250

Chapter 7
Obtaining Information from the Plotter

What You'll Learn in This Chapter

Up to this time we have mainly been concerned with sending information or data to the plotter. Sometimes, however, we want to know something about the plotter, its current pen position, its status, whether an error has occurred, or what capabilities the plotter has. In this chapter you will learn about most of the plotter's output instructions. The output P1 and P2, the output window, and the output hard-clip limits instructions are discussed in Chapter 2 and the output digitized point instruction is discussed in Chapter 6. All other output instructions are discussed in this chapter. The timing of output depends on your interface (HP-IB or RS-232-C). Before using the output instructions, you should have read the notes below and the appropriate interfacing chapter in this manual.

HP-GL Instructions Covered

OA The Output Actual Position and Pen Status Instruction
OC The Output Commanded Position and Pen Status Instruction
OE The Output Error Instruction
OF The Output Factors Instruction
OI The Output Identification Instruction
OO The Output Options Instruction
OS The Output Status Instruction

Terms You Should Understand

Output Terminator — denoted in this manual as TERM — the ASCII character or characters sent by the plotter at the end of a plotter response to an output instruction. With an HP-IB interface, the two characters, carriage return and line feed, are the output terminator. With an RS-232-C interface, the output terminator is a carriage return, unless modified by an ESC . M command.

A Brief Word about Plotter Output

There are slight differences in the timing of output when the plotter is used with the HP-IB or RS-232-C interfaces. Read the paragraph below which pertains to your system.

Notes for an HP-IB User

When the 7475 has an HP-IB interface, the terminator for an output statement, denoted TERM, is a carriage return followed by a line feed.

The output instructions in this chapter should not be used when the plotter is in listen-only mode since the plotter in listen-only mode cannot output anything. Output instructions will be ignored by the plotter so the computer will get no response to its read statement, and, typically, the program will halt.

A plotter with an HP-IB interface will respond only when the computer sends a read instruction (the plotter is instructed to talk). Therefore, a read statement should directly follow any output instruction. When a second output instruction is received before data from the first instruction has been read, the new data overwrites the old data and the old data is lost. Refer to Chapter 9 for more information.

Notes for an RS-232-C User

With an RS-232-C interface, the 7475's terminator for an output statement, denoted TERM, is a carriage return, unless the terminator is modified by an ESC . M instruction. As soon as an output instruction has been parsed by the plotter, output occurs according to the handshake protocol established by the ESC . M and ESC . N instructions. Use of turnaround delays, intercharacter delays, and an output initiator should be specified as necessary to assure that output will not be lost because the computer is not prepared to receive it. The information necessary to assure this should be contained in the documentation for your computer. Refer to Chapter 10 of this manual for more information.

The Output Actual Position and Pen Status Instruction, OA

DESCRIPTION The output actual position and pen status instruction, OA, is used to output the X- and Y-coordinates and pen status (up or down) associated with the actual pen position.

USES This instruction can be used to determine the pen's current position in plotter units. You might use that information to position a label or figure, or determine the parameters of some desired window.

SYNTAX *OA* terminator

EXPLANATION Output is always in plotter units.

No parameters are used.

The pen position and status are output to the computer as integers in ASCII in the form:

X,Y,P TERM

where X is always the X-coordinate in plotter units,
Y is always the Y-coordinate in plotter units,
P is the pen status (0 = pen up, 1 = pen down), and
TERM is the output terminator for the interface installed.

The ranges of the X- and Y-coordinates are the hard-clip limits determined by the setting of the paper switches.

Hard-clip Limits

Paper Size	Hard-clip Limits	
	X-axis	**Y-axis**
A	$0 \leqslant X \leqslant 10\,365$	$0 \leqslant Y \leqslant 7962$
B	$0 \leqslant X \leqslant 16\,640$	$0 \leqslant Y \leqslant 10\,365$
A4	$0 \leqslant X \leqslant 11\,040$	$0 \leqslant Y \leqslant 7721$
A3	$0 \leqslant X \leqslant 16\,158$	$0 \leqslant Y \leqslant 11\,040$

No positive sign is output.

The Output Commanded Position and Pen Status Instruction, OC

DESCRIPTION The output commanded position and pen status instruction, OC, is used to output the X- and Y-coordinates and pen status (up or down) associated with the last valid pen position instruction.

USES This instruction can be used to determine the pen's last valid commanded position in plotter units or user units depending on whether scaling is off or on. You might use that information to position a label or figure, or determine the parameters of an instruction which moved the pen to the limits of some window.

SYNTAX *OC* terminator

EXPLANATION Output is in decimal format, in user units when scaling is in effect, and in plotter units when scaling is off.

No parameters are used.

The pen position and status are output to the computer as decimal numbers in ASCII in the form:

X,Y,P TERM

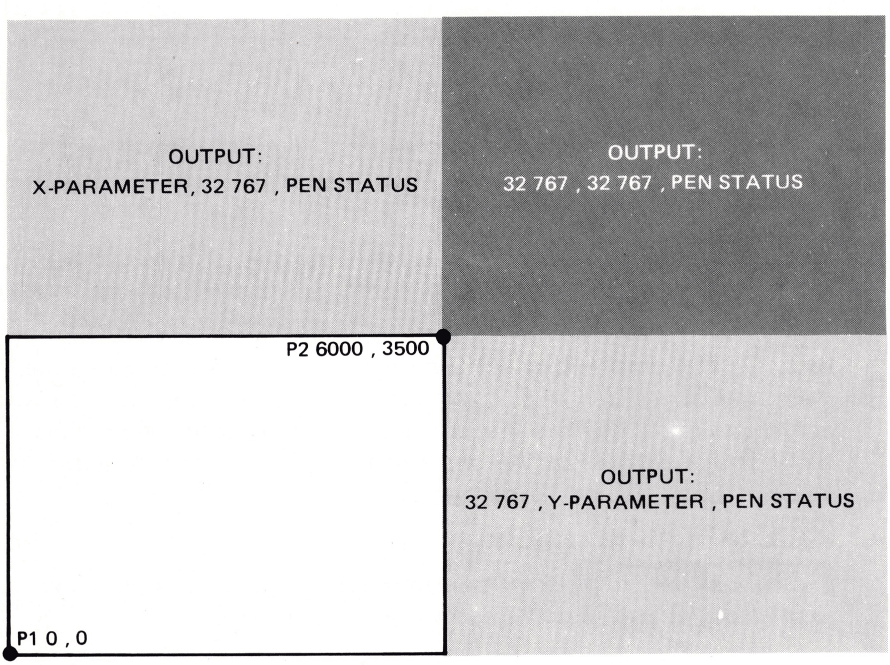
where X is always the X-coordinate in plotter units or user units,
Y is always the Y-coordinate in plotter units or user units,
P is the pen status (0 = pen up, 1 = pen down), and
TERM is the output terminator for the interface installed.

When scaling is off, X- and Y-coordinates are in plotter units. When scaling is on, X- and Y-coordinates are in user units. Ranges of the X-and Y-coordinates are −32 768 to 32 767 whether scaling is on or off.

When the commanded pen position is such that its user unit value would be less than −32 768 or greater than 32 767, the output may not represent the true pen position. If the plotter were scaled with the given instructions as shown in the following illustration, all points in the lightly shaded areas will have one coordinate as 32 767, the largest number the plotter can output. All points in the darker shaded area will have both coordinates as 32 767. One way to access this area is with the AA instruction.

Instructions executed:

```
"IP 0,0,6000,3500; SC 0,32767,0,32767;"
```



The Output Error Instruction, OE

DESCRIPTION The output error instruction, OE, is used to output the decimal equivalent of the first HP-GL error (if any).

USES This instruction can be used to determine the type of the first error. It is useful when debugging programs or to determine if all data or instructions were accepted by the plotter.

SYNTAX *OE* terminator

EXPLANATION No parameters are used.

When an OE instruction is received, the plotter converts the first HP-GL error to a positive integer in ASCII, which is output in the form:

error number TERM

The error number is defined as follows:

Error Number	Meaning
0	No error
1	Instruction not recognized
2	Wrong number of parameters
3	Out-of-range parameters
4	Not used
5	Unknown character set
6	Position overflow
7	Not used
8	Vector received while pinch wheels raised

TERM is the output terminator for the interface installed.

In an HP-IB system after the carriage return has been sent, and in an RS-232-C system after the output is complete, bit position 5 of the status byte is cleared (if set), and the **ERROR** LED (if lit) is turned off (unless there is an RS-232-C error which has not been cleared by an ESC.E instruction).

You should note that anytime the plotter receives an unpaired alphabetic character, error 1 will be set. Thus, an alphabetic parameter or three alphabetic characters in a row will generate error 1. When you encounter error 1, look for a misplaced alphabetic character.

Once your plotting programs are debugged, you may want to remove most output error instructions from your program to reduce your computer's I/O operations and maximize plotting speed.

The Output Factors Instruction, OF

DESCRIPTION The output factors instruction, OF, is used to output the number of plotter units per millimetre in each axis.

USES This instruction enables the plotter to be used with software which must know the size of a plotter unit.

SYNTAX *OF* terminator

EXPLANATION No parameters are used.

The plotter will always output the following:

40 , 40 TERM

These factors indicate that there are approximately 40.2 plotter units per millimetre in the X-axis and in the Y-axis (0.025 mm/plotter unit). TERM is the output terminator for the interface installed.

The Output Identification Instruction, OI

DESCRIPTION The output identification instruction, OI, is used to output a plotter identifier.

USES This instruction is especially useful in a remote operating environment to determine which model plotter is on-line.

SYNTAX *OI* terminator

EXPLANATION No parameters are used.

The plotter will always output the following character string:

7475A TERM

TERM is the output terminator for the interface installed.

The Output Options Instruction, OO

DESCRIPTION The output options instruction, OO, is used to output eight option parameters.

USES This instruction is especially useful in a remote operating environment to determine which options are available in the plotter which is on-line.

SYNTAX *OO* terminator

EXPLANATION No parameters are used.

The plotter will always output the appropriate combination of eight integers in ASCII, separated by commas. The options included in the plotter are indicated by a 1 as defined below.

0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 TERM

Indicates arcs and circle instructions are included.

Indicates pen select capability is included.

TERM is the output terminator for the interface installed.

The Output Status Instruction, OS

DESCRIPTION The output status instruction, OS, is used to output the decimal equivalent of the status byte.

USES This instruction is useful in debugging operations and in digitizing applications.

SYNTAX *OS* terminator

EXPLANATION No parameters are used.

Upon receipt of the OS instruction, the internal eight-bit status byte is converted to an integer between 0 and 255. Output is in ASCII in the form:

status TERM

The status bits are defined as follows:

Bit Value	Bit Position	Meaning
1	0	Pen down.
2	1	P1 or P2 changed; cleared by reading output of OP in HP-IB system or by actual output of P1,P2 in RS-232-C system, or by IN instruction.
4	2	Digitized point available; cleared by reading digitized value in HP-IB system or by output of point in RS-232-C system, or by IN instruction.
8	3	Initialized; cleared by reading OS output in HP-IB system or by output of the status byte in RS-232-C system.
16	4	Ready for data; pinch wheels down.
32	5	Error; cleared by reading OE output in HP-IB system or by output of the error in RS-232-C system, or by IN instruction.
64	6	Require service message set (always 0 for OS ; 0 or 1 for HP-IB serial poll):
128	7	Not used

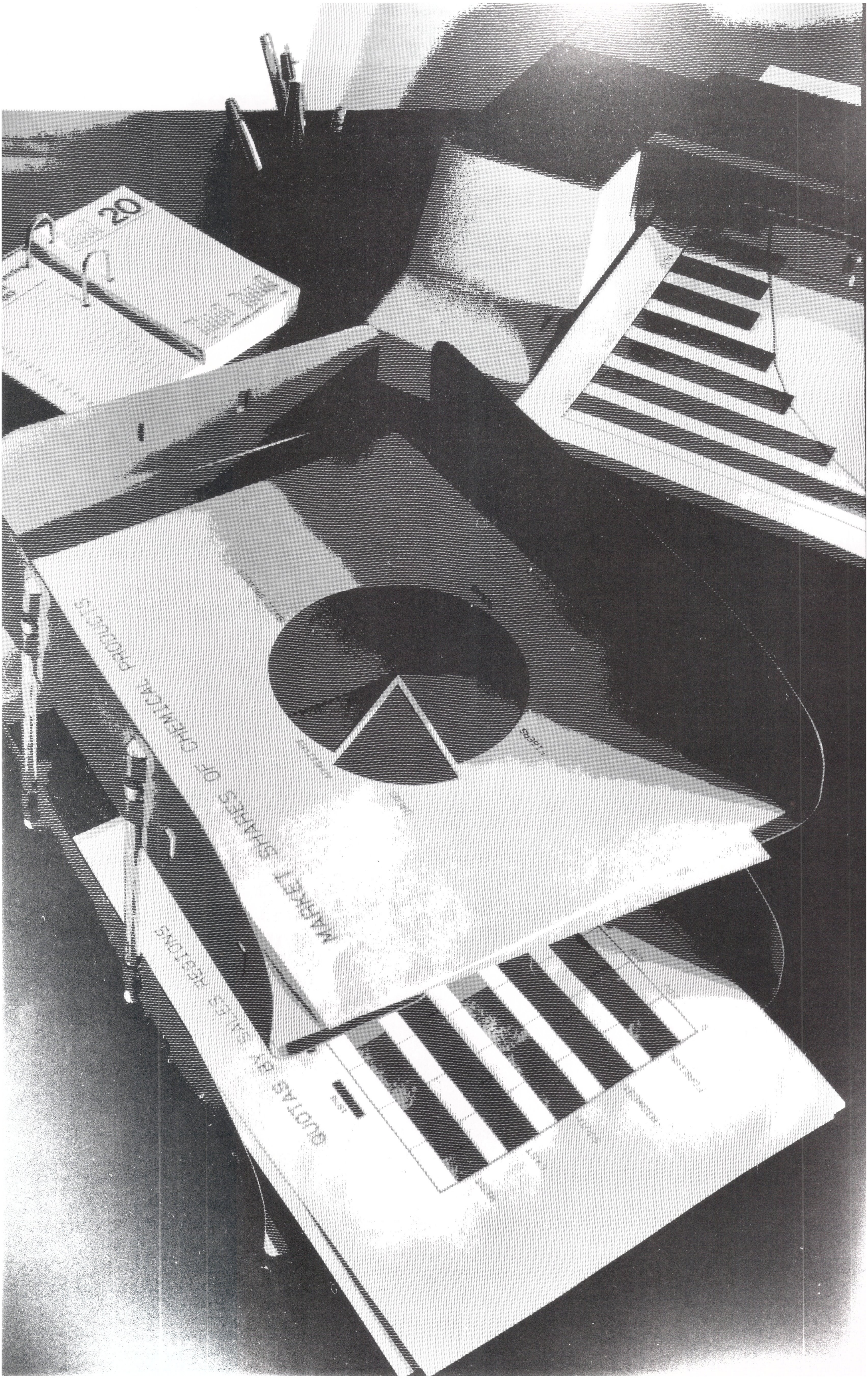
Upon power up, the status is decimal 24, the sum of 8 (initialized) and 16 (ready for data). Upon output of the status byte after an OS instruction, bit position 3 is cleared.

Summary of Output Response Types

The following table shows the number and type of items in the response to each HP-GL output instruction. The table includes output instructions explained in Chapters 2 and 6 as well as in this chapter. This table will be helpful when programming in languages such as FORTRAN which require you to specify the type of and number of digits in a variable.

Instruction	Number of Parameters Returned*	Type and Range
OA	3	integers, all $\leqslant$ 5 digits
OC	3	maximum 5 digits in integer portion, maximum 4 digits in fractional portion (sign and decimal point optional)
OD	3	integers, all $\leqslant$ 5 digits
OE	1	integer, 1 digit
OF	2	integers, 2 digits each
OI	1	5-character string
OO	8	integers, 1 digit each
OP	4	integers, all $\leqslant$ 5 digits
OS	1	integer, $\leqslant$ 3 digits
OW	4	integers, all $\leqslant$ 5 digits

*In addition to these parameters, the output terminator TERM is always sent at the end of output, and commas are sent to separate parameters.


MARKET SHARES OF CHEMICAL PRODUCTS
QUOTAS BY SALES REGIONS
20

Chapter 8
Putting the Instructions to Work

What You'll Learn in This Chapter

In this chapter you'll learn how to put instructions together to develop a plot. The following examples are designed to show you how to integrate many instructions into a complete program, how data might be handled, and how subroutines are used to program a task that is common to many plots and could be used in several programs.

Remember that these programs are written in Microsoft® BASIC. They use techniques such as FOR . . . NEXT loops and subroutines to read data and draw plots. If necessary, check your computer documentation for the correct methods of implementing these techniques.

The first program draws a line chart, one of the most common types of plots. You can use line charts to plot almost any kind of data — sales data, factory output, sales volume, data from laboratory experiments, population trends, etc. The concepts of plotting and labeling demonstrated here can be used in almost any application.

The second program draws a stacked bar chart; the third program draws a pie chart. The sales data are differentiated in bars or wedges by solid fill, cross-hatching and parallel hatching. The programs demonstrate how to define fill types and how to fill and outline rectangles and wedges.

The first program is explained in detail, and is organized to show you how to develop a program. The second two programs are explained more briefly, because the concepts of developing these programs are similar to developing the line chart.

NOTE: Some computers use an Xon-Xoff handshake to prevent buffer overflow and data loss. ■

To set up an Xon-Xoff handshake, insert the following lines in your program after the configuration statement. (For more information, refer to the ESC . I and ESC . N instructions in Chapter 9.)

```
PRINT #1, CHR$(27);".I 81;;17:"
PRINT #1, CHR$(27);".N;19:"
```

Line Chart

For this line chart, you will scale, draw, and label an X- and Y-axis and plot 1985 sales by region. The following paragraphs develop segments of the program in a logical sequence. The complete plot and program are shown later in the section titled Program Listing.

Setup and Scaling

For emphasis and readability, you should draw the title and important data with wide pens. Narrow pens are usually sufficient for axes and labels. For this line chart, the suggested pen order for the carousel is:

1 = black, P.3	5 = green, P.3
2 = black, P.7	6 = aqua, P.3
3 = red, P.3	7 = unused
4 = blue, P.3	8 = unused

Begin your program with the appropriate configuration statement for your computer. Then, using the IN or DF instruction, set the plotter to known conditions and cancel any parameters that may have been set in a previous program. IN is used here to be sure *all* conditions (such as P1/P2 settings) are set to a default state.

Select a pen (SP 1;) and establish scaling points for this plot. The parameters of the IP instruction determine the location of the scaling points, P1 and P2. The location of these points provides a convenient area for the scale, which is assigned in the scaling statement SC 1,12,0,150;. Since this chart shows one year's sales by month, the X-axis (commonly representing time) is scaled from 1 to 12. The Y-axis is scaled in thousands from 0 to 150 so that all sales data will fall inside this range. Labels and titles will be placed outside this area.

You will either need to know the range of your data or be willing to try some plots with different scales to determine what your scale statement should be. Thousands or millions of dollars are common scales.

Once the scale is established, draw a frame for the data area. Here PU 1,0; moves the pen to the first point with the pen up. The pen is then lowered and connects the four corners.

The first three program lines to accomplish the above are:

```
20   PRINT #1, "IN;SP1;IP1250,750,9250,6250;"
30   PRINT #1, "SC1,12,0,150;"
40   PRINT #1, "PU1,0;PD12,0,12,150,1,150,1,0;PU;"
```

The Axes and Their Labels

You are now ready to draw and label the axes. The absolute label size instruction, SI 0.2, 0.3;, creates characters slightly larger than the default character size. The tick length is established by the instruction TL 1.5,0;. The resulting ticks will be 1.5% of the horizontal or vertical distances between scaling points.

This program uses a FOR . . . NEXT loop to draw the axes. For the X-axis, let X range from 1 to 12 to represent 12 months of data. The loop does four things: moves to the integer location on the X-axis, draws a tick mark, establishes the label origin, and draws the label. Note that the X-parameter of the plotting instruction is a variable. If you do not know how to send a variable to the plotter, consult your computer's documentation and Plotting with Variables in Chapter 3. Since the XT instruction draws a tick whether the current pen status is up or down, be sure the pen is up to avoid unwanted lines between the ticks, labels, and axis.

Place the labels in a DATA statement in order to use the looping technique for labeling axes. (At some point, you might want to access data for the latest 12 months. If your data were stored with a data code, you could use a similar technique to read the labels and data from a file and properly label your chart for the data you were then plotting.) Then access the labels with a string variable in the LB instruction. Refer to Labeling with Variables in Chapter 5 for hints on sending variables in labels.

To position the labels, the program uses the CP instruction to center the label under the tick. By moving one-third character space back and one line down, the single character label is centered under the tick with enough space to be easily read. Finally, the axis title, Calendar Month, is centered and drawn under the axis.

The following lines contain the statements that perform the functions just described.

```
50   PRINT #1, "SI.2,.3;TL1.5,0;"
60   FOR X = 1 TO 12
70     PRINT #1, "PA";X;",0;XT;"
80     READ A$
90     PRINT #1, "CP-.33,-1;LB"+A$+CHR$(3)
100  NEXT X
110  PRINT #1, "PA6.5,0;CP-7,-2.5;"
120  PRINT #1, "LBCalendar Month"+CHR$(3)

500  DATA "J","F","M","A","M","J"
510  DATA "J","A","S","O","N","D"
```

The Y-axis is created in a similar manner, except that the program uses the loop's index for the label value and two different CP instructions for labels of three digits and labels of less than three digits.

The lines which draw the Y-axis and label it follow.

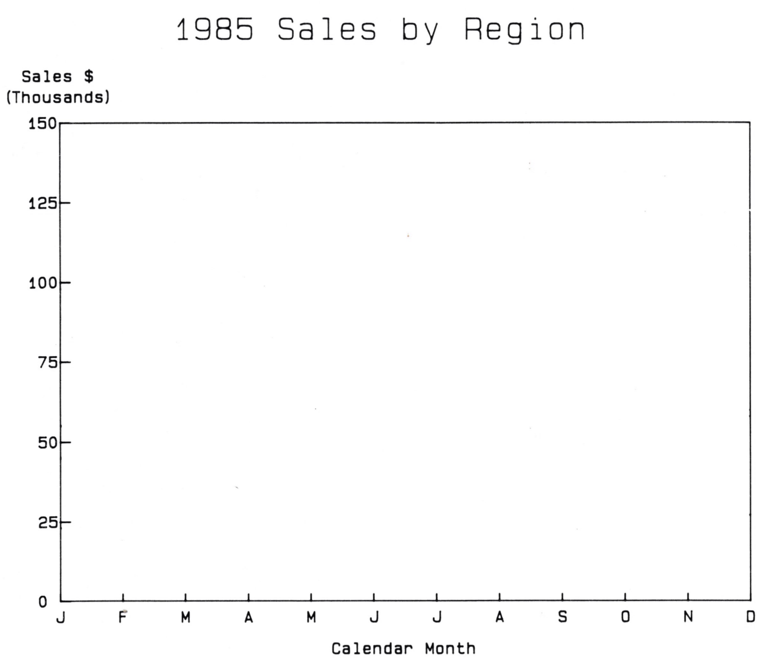
```
130  FOR Y=0 TO 150 STEP 25
140    PRINT #1, "PA1,";Y;";YT;"
150    IF Y<100 THEN PRINT #1, "CP-3,-.25;"
160    IF Y>99 THEN PRINT #1, "CP-4,-.25;"
170    PRINT #1, "LB";Y;CHR$(3)
180  NEXT Y
190  PRINT #1, "PA1,150;CP-3.5,2;"
200  PRINT #1, "LBSales $"+CHR$(3)+"CP-9,-1;"
210  PRINT #1, "LB(Thousands)"+CHR$(3)
```

Change to a wide pen to plot the title. Next, move to the top center of the chart, increase the character size, and label the chart title.

The program lines that title the chart are:

```
220  PRINT #1, "SP2;PA6,150;SI.4,.6;CP-9.5,2;"
230  PRINT #1, "LB1985 Sales by Region"+CHR$(3)
```

Here's what the chart looks like so far.



Plotting the Data

You are now ready to draw lines. The first data line is drawn with parameters included when the program was written. Therefore, if the data changes, it will be necessary to change the plot instructions in the program.

The first line is drawn with pen 6 using the default solid line type. After drawing the line, the pen moves (up) to an area appropriate for labeling. After the character size is changed to match that used to label the axes, the plotter labels "South America." Actually, each of the data line's labels were inserted near the end of the creation process and involved trial and error to achieve satisfactory placement. Each label is drawn after each line of data is plotted.

The program lines which plot the lowest line and the corresponding label are:

```
240  PRINT #1, "SP6;LT;PA1,23;PD2,25,3,18,4,22;"
250  PRINT #1, "PD5,23,6,27,7,27,8,25,9,24,10,28;"
260  PRINT #1, "PD11,27,12,27,;PU3.6,16;"
270  PRINT #1, "SI.2,.3;LBSouth America"+CHR$(3)
```

The program plots the three remaining lines from data read at execution time using nested FOR . . . NEXT loops and a READ statement. You can use this technique to plot a chart that will be replotted often with new data. If the necessary file statements were added, the data could be on a tape or disk file instead of in a DATA statement as shown here.

The first FOR . . . NEXT loop beginning in line 280 runs 3 times, once for each of the remaining data lines. With each loop sequence, a new pen color and line type (3, 4, 5) are selected in line 290.

In line 300 the second FOR . . . NEXT loop begins. It runs 12 times to read each of the 12 values in the DATA statement and draw to each point. As with the first data line, the corresponding label is drawn (lines 340–360) after each line is plotted.

NOTE: Since this program uses variables as plot parameters, be sure they are sent to the plotter with a valid separator between them. Here, a comma has been inserted between variables to *ensure* that they are separated, even though many systems do not require this. Computers often send a leading and/or trailing blank space, or allow for a sign space before numeric variables. The plotter will treat a blank, comma, or a plus or minus sign as a separator between numeric parameters. Know your computer before sending variables with plot instructions. ■

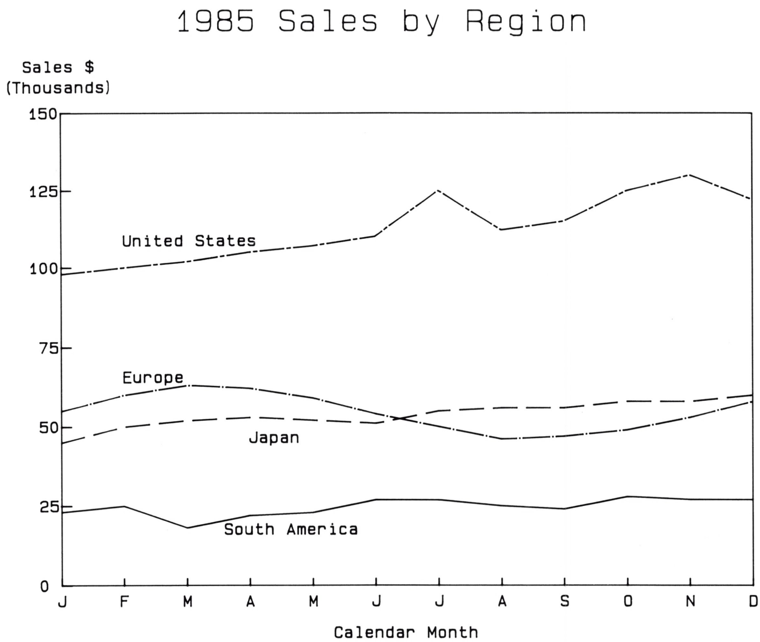
The loops that draw the remaining three chart lines and the corresponding data statements follow. Although 340–360 are each printed on

two lines to fit on this page, send them to the plotter as one continuous string.

```
280  FOR I=1 TO 3
290    PRINT #1, "SP";I+2;";LT";I+2;";"
300    FOR X = 1 TO 12
310      READ Y
320      PRINT #1, "PA";X;",";Y;"PD;"
330    NEXT X
340    IF I=1 THEN PRINT #1, "PU4,45;LBJapan"
                             +CHR$(3)
350    IF I=2 THEN PRINT #1, "PU2,64;LBEurope"
                             +CHR$(3)
360    IF I=3 THEN PRINT #1, "PU2,107;LBUnited
                             States"+CHR$(3)
370  NEXT I
380  PRINT #1, "SP0;"
500  DATA "J","F","M","A","M","J"
510  DATA "J","A","S","O","N","D"
520  DATA 45,50,52,53,52,51,55,56,56,58,58,60
530  DATA 55,60,63,62,59,54,50,46,47,49,53,58
540  DATA 98,100,102,105,107
550  DATA 110,125,112,115
560  DATA 125,130,122,0,0
570  END
```

Program Listing

A reduced version of the plot is shown next, followed by a complete listing of the program. Line 10 must include the proper configuration instructions necessary to establish interface conditions. You might need to make changes for your computer's BASIC. Or, you can use another programming language and send the HP-GL instructions using that language's output and looping techniques.




```
10  OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20   PRINT #1, "IN;SP1;IP1250,750,9250,6250;"
30   PRINT #1, "SC1,12,0,150;"
40   PRINT #1, "PU1,0;PD12,0,12,150,1,150,1,0;PU;"
50   PRINT #1, "SI.2,.3;TL1.5,0;"
60   FOR X = 1 TO 12
70     PRINT #1, "PA";X;",0;XT;"
80     READ A$
90     PRINT #1, "CP-.33,-1;LB"+A$+CHR$(3)
100  NEXT X
110  PRINT #1, "PA6.5,0;CP-7,-2.5;"
120  PRINT #1, "LBCalendar Month"+CHR$(3)
130  FOR Y=0 TO 150 STEP 25
140    PRINT #1, "PA1,";Y;";YT;"
150    IF Y<100 THEN PRINT #1, "CP-3,-.25;"
160    IF Y>99 THEN PRINT #1, "CP-4,-.25;"
170    PRINT #1, "LB";Y;CHR$(3)
180  NEXT Y
190  PRINT #1, "PA1,150;CP-3.5,2;"
200  PRINT #1, "LBSales $"+CHR$(3)+"CP-9,-1;"
210  PRINT #1, "LB(Thousands)"+CHR$(3)
220  PRINT #1, "SP2;PA6,150;SI.4,.6;CP-9.5,2;"
230  PRINT #1, "LB1985 Sales by Region"+CHR$(3)
240  PRINT #1, "SP6;LT;PA1,23;PD2,25,3,18,4,22;"
```

```
250  PRINT #1, "PD5,23,6,27,7,27,8,25,9,24,10,28;"
260  PRINT #1, "PD11,27,12,27,;PU3.6,16;"
270  PRINT #1, "SI.2,.3;LBSouth America"+CHR$(3)
280  FOR I=1 TO 3
290    PRINT #1, "SP";I+2;";LT";I+2;";"
300    FOR X = 1 TO 12
310      READ Y
320      PRINT #1, "PA";X;",";Y;"PD;"
330    NEXT X
340    IF I=1 THEN PRINT #1, "PU4,45;LBJapan"
                              +CHR$(3)
350    IF I=2 THEN PRINT #1, "PU2,64;LBEurope"
                              +CHR$(3)
360    IF I=3 THEN PRINT #1, "PU2,107;LBUnited
                              States"+CHR$(3)
370  NEXT I
380  PRINT #1, "SP0;"
500  DATA "J","F","M","A","M","J"
510  DATA "J","A","S","O","N","D"
520  DATA 45,50,52,53,52,51,55,56,56,58,58,60
530  DATA 55,60,63,62,59,54,50,46,47,49,53,58
540  DATA 98,100,102,105,107
550  DATA 110,125,112,115
560  DATA 125,130,122,0,0
570  END
```

Bar Graphs and Pie Charts

Filling and Hatching

Two kinds of area fill are commonly used in bar graphs and pie charts; solid fill and hatching. Solid fill totally covers the area with color, whereas hatching fills the area with evenly spaced parallel lines. If there are lines in two directions at 90-degree angles, we call the hatching crosshatching. Sometimes a graph will have both narrow and wide hatching or crosshatching, the wide hatching having more space between the lines than the narrow.

Producing a Bar Graph

Scaling the Axes

In the following bar graph titled "Sales Volume by Region," we are plotting sales over a three-year period. For readability, the X-axis is scaled to provide a comfortable margin of space before and after each bar. The Y-axis is scaled from 0 to 500 to represent sales in thousands of dollars.

Plotting the Title

The title and axes are drawn with a wide pen (stored in stall two) for emphasis. The title is drawn first, with characters that are a little more than twice the default size. All labels are centered and offset slightly from the data area with the CP instruction.

Labeling the Axes

The bars on the X-axis are labeled without tick marks, using a narrow pen and characters that are slightly larger than default size. The Y-axis is labeled with tick marks and an extra label to show the scaling used (K$).

Labeling the Bar Segments

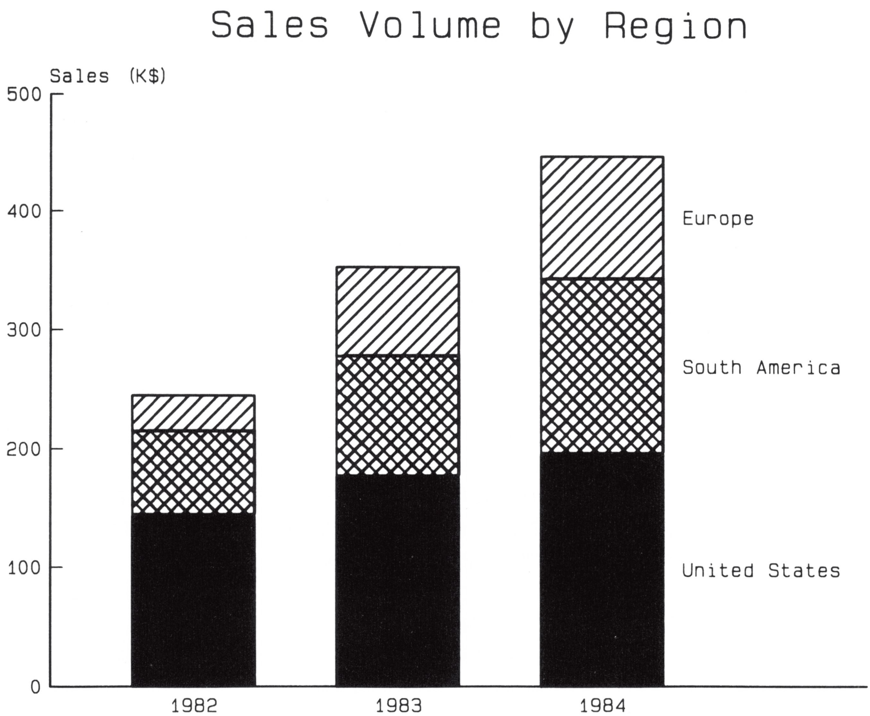
The data for labeling each bar segment is input using read and data statements. This approach allows easy modification of the label data. Each segment label is centered next to the rightmost bar by computing a Y-axis position that is equal to the height of the prior segments plus one-half the height of the current segment.

Filling and Edging Each Segment

The data for each bar segment is stored in a three by three array. Each array element contains the height of a segment with respect to the Y-axis scaling. The bars are drawn from bottom to top using the FT, RA, and EA instructions to define, fill, and edge each stacked rectangular segment. Wide pens are used in stalls 3, 4, and 5 of the carousel for filling the bars.

Completion of Bar Graph

At the completion of the program, the scaling points are reset to their default location, the pen is raised and put away, and the finished plot is presented for viewing.




```
10     REM Generalized bar chart
20     REM
30     REM  Place chart and label data in arrays
40 REM  with lower bounds of 1.
50      OPTION BASE 1
60      DIM L$(80),B(3,3)
70     REM
80      DATA United States,South America,Europe
90      DATA 144,177,196,71,101,147,30,75,104
100     READ L$(1),L$(2),L$(3)
110     FOR I = 1 TO 3
120       FOR J = 1 TO 3
130         READ B(I,J)
140       NEXT J
150     NEXT I
160    REM Configuration statement
170 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
180 REM  Initialize plotter; set scaling points P1,P2
190 REM  Scale axes
200 PRINT #1, "IN;IP1000,1000,9000,6750;"
210 PRINT #1, "SC1981,1985,0,500;"
```

(Program listing continued)

```
220 REM Label title using thick pen
230 REM then draw axes
240 PRINT #1, "PU;SP2;PA1983.3,500;SI.4,.6;CP-10,1.2;"
250 PRINT #1, "LBSales Volume by Region"+CHR$(3)
260 PRINT #1, "PU;PA1981.3,500;PD;PA1981.3,0,1985.3,0;"
270 REM Select narrow pen; reset character size.
280 REM Set tick length; then label axis.
290  PRINT #1, "PU;SP1;SI.2,.3;TL1.5,0;"
300  FOR X=1982 TO 1984
310    PRINT #1, "PA";X;",0;CP-2.8,-1;LB";X;CHR$(3)
320  NEXT X
330 REM Add ticks and then labels to the Y-axis.
340  FOR Y= 0 TO 500 STEP 100
350    PRINT #1, "PA 1981.3,";Y;";YT;"
360    IF Y=0 THEN PRINT #1, "CP-2.5,-.25;"
370    IF Y<>0 THEN PRINT #1, "CP-4.5,-.25;"
380    PRINT #1, "LB";Y;CHR$(3)
390  NEXT Y
400  PRINT #1, "PA1981.3,510;LBSales (K$)"+CHR$(3)
410 REM Center segment labels using prior height
420 REM plus one-half current height.
430  FOR I=1 TO 3
440     Y=0
450       FOR J=1 TO I-1
460         Y=Y+B(J,3)
470.      NEXT J
480     Y=Y+B(I,3)/2
490     PRINT #1, "PA1984.4,";Y;";"
500     PRINT #1, "CP0,-.25;LB"+L$(I)+CHR$(3)
510   NEXT I
520  REM Draw and fill each bar using wide pens.
530   FOR I=1 TO 3
540     PRINT #1, "SP";I+2;";PT.7;"
550     K=1
560     FOR X=1982 TO 1984
570       Y1=0
580  REM Compute Y-axis start point
590  REM for each bar segment.
600      FOR J=1 TO I-1
610        Y1=Y1+B(J,K)
620      NEXT J
630  REM Compute Y-axis end point
640  REM for each bar segment.
650      Y2=Y1+B(J,K)
660      K=K+1
670  REM Select fill type.
```

```
680       IF I=1 THEN PRINT #1, "FT1;"
690       IF I=2 THEN PRINT #1, "FT4,.05,45;"
700       IF I=3 THEN PRINT #1, "FT3,.05,45;"
710   REM Move to start point for segment;
720   REM then fill and outline defined bar area.
730       PRINT #1, "PA";X-.3;",";Y1;";"
740       PRINT #1, "RA";X+.3;",";Y2;";"
750       PRINT #1, "EA";X+.3;",";Y2;";"
760     NEXT X
770   NEXT I
780   REM Put pen away and end program.
790    PRINT #1, "SP0;"
800   END
```

Producing a Pie Chart

Overview

In the following pie chart titled "Sales Dollar Distribution," we are plotting the distribution of sales dollars among four groups: R&D, Administration, Marketing, and Manufacturing. Pie charts easily convey information concerning parts of a whole. Since we can easily break up the whole sales-dollar-distribution picture into four parts, the pie chart is an appropriate choice for presenting this information. For ease of understanding, pie charts ought not to be broken into less than three or more than six parts.

Inputting the Data

In the loop beginning at line 110, we input the start, mid-point, and stop angle for each segment using read and data statements. In line 155, the pie segment labels are read into arrays.

Plotting the Title

So that the pie chart will be centered, P1 and P2 are repositioned with the IP instruction and the plotting area is scaled so that 0,0 is near the center of the page. The title is centered with the CP instruction, the character size is set to 0.4 cm wide and 0.6 cm high with the SI instruction, and a wide pen is selected with the SP instruction.

Plotting the Labels

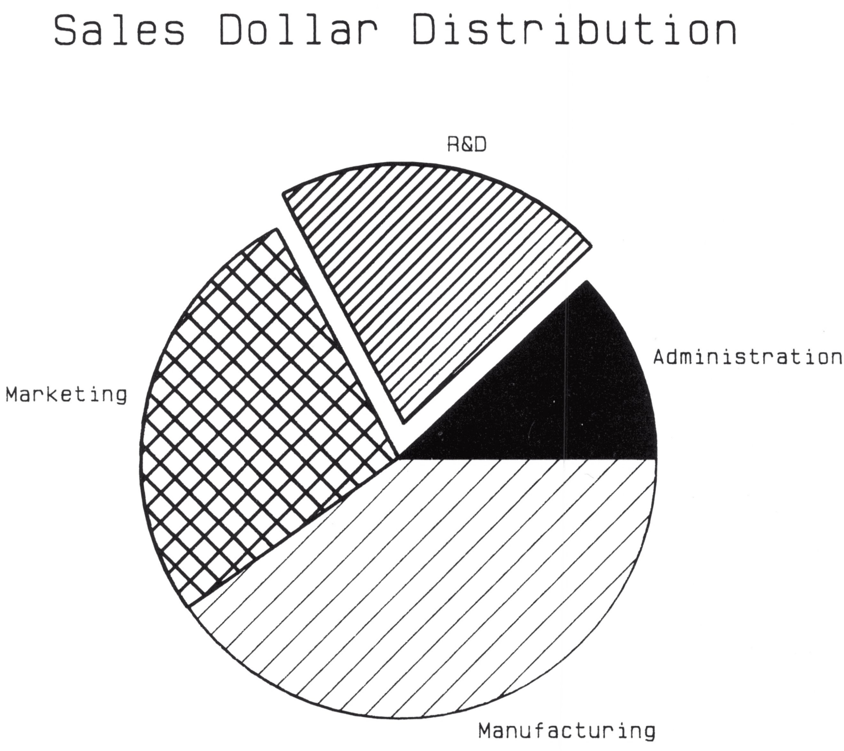
Lines 220 through 340 label each segment using the data previously stored in line 90. For readability, the labels are drawn with a narrow pen and with a character size that is smaller than the one used for the title. The labels are centered and offset from the appropriate segment with the CP instruction. We increase the distance used to position the label for segment 2, since the second segment is exploded.

Filling and Edging Each Segment

In the next loop beginning at line 360, a wide pen and a fill type are selected for each segment with the SP and FT instructions. Lines 400 and 410 adjust the center point for the exploded segment. The sweep angle for each segment is then computed (sweep angle = stop angle − start angle), and the segments are filled and outlined with the WG and EW instructions.

Completion of Pie Chart

At the completion of the program, the scaling points are reset to their default location, the pen is raised and put away, and the finished plot is presented for viewing.

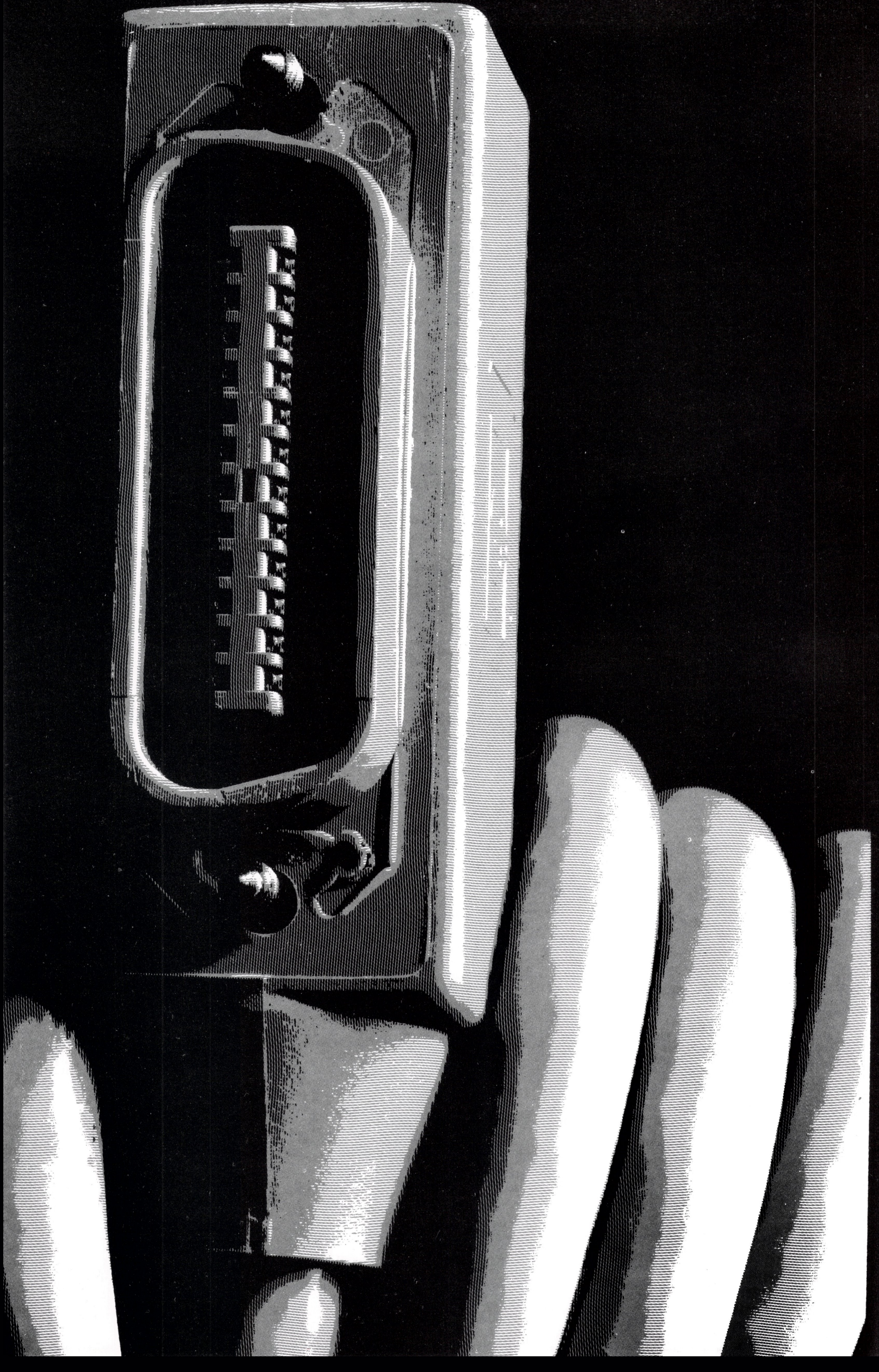



```
10  OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
20  REM Pie Chart
30  REM Place chart and label data in arrays
40  REM with lower bounds of 1
50  OPTION BASE 1
60  DIM W$(4),P(4,3)
70  DATA 0,21.5,43,43,80.5,118,118,166.5
```

(Program listing continued)

```
80  DATA 215,215,287,360
90  DATA Administration, R&D,Marketing,Manufacturing
100 DATA Manufacturing
110 REM Read start, mid-point, and stop angle
120 REM for each segment.  Then read labels.
130 FOR I = 1 TO 4
140   FOR J = 1 TO 3
150      READ P(I,J)
160   NEXT J
170 NEXT I
180 READ W$(1),W$(2),W$(3),W$(4)
190 REM Initialize plotter; set scaling points
200 REM P1 and P2; scale the chart.
210 PRINT #1,"IN;IP2250,500,8250,7100;"
220 PRINT #1,"SC-10,10,-10,12;"
230 REM Label title using wide pen, large letters.
240 PRINT #1,"SP2;PA0,12;SI.4,.6;CP-12.34,0;"
250 PRINT #1,"LBSales Dollar Distribution"+CHR$(3)
260 REM Label each wedge using narrow pen and
270 REM small letters.
280 PRINT #1,"SP1;SI.2,.3;"
290 REM Set PI variable to convert from
300 REM radians to degrees.
310 PI=3.141593
320 FOR I = 1 TO 4
330 REM Move to center arc.
340   R = 8
350   IF I=2 THEN R=9
360   X=R*COS(P(I,2)*(PI/180))
370   Y=R*SIN(P(I,2)*(PI/180))
380   PRINT #1,"PA";X;",";Y;";"
390   REM Determine label origin ; draw label
400   L = LEN  (W$(I))
410   IF I = 1 THEN PRINT #1,"CP0,-.25;"
420   IF I = 3 THEN PRINT #1,"CP";-L;"-.25;"
430   IF I = 4 THEN PRINT #1,"CP0,-.5;"
440   PRINT #1,"LB"+W$(I)+CHR$(3)
450 NEXT I
460 REM Draw and fill the wedges using pens 3-6
470 FOR I = 1 TO 4
480    PRINT #1,"SP";I+2;";PT.7;"
490    X=0
500    Y=0
510    IF I = 2 THEN X=COS(P(I,2)*(PI/180))
520    IF I = 2 THEN Y=SIN(P(I,2)*(PI/180))
530    IF I = 1 THEN PRINT #1,"FT1;"
```

```
540     IF I = 2 THEN PRINT #1,"FT3,.3,45;"
550     IF I = 3 THEN PRINT #1,"FT4,.6,45;"
560     IF I = 4 THEN PRINT #1,"FT3,.6,45;"
570     REM Compute the sweep angle
580     S=P(I,3)-P(I,1)
590     REM Fill the wedge.
600     PRINT #1,"PA";X;",";Y;";"
610     PRINT #1,"WG7.5,";P(I,1);",";S;";"
620     REM Outline the wedge.
630     PRINT #1,"EW7.5,";P(I,1);",";S;";"
640 NEXT I
650 REM Put pen away and end program
660 PRINT #1,"SP0;"
670 END
```



Chapter 9
HP-IB Interfacing

What You'll Learn in This Chapter

This chapter is only for 7475 owners with an HP-IB interface. HP 7475s with Option 002 have an HP-IB interface.

In this chapter you'll learn how to operate your plotter when it is connected to a computer using the Hewlett-Packard Interface Bus (HP-IB), which conforms to ANSI/IEEE 488-1978 specifications. This chapter defines the 7475's implementation of the bus. Also included are addressing the 7475, the listen-only mode, reaction to bus clear commands, serial and parallel polling, addressing the 7475 as a talker or listener, and examples of sending and receiving data using a variety of computers.

This chapter assumes you have a working knowledge of the HP-IB; however, if you wish to refresh your memory on HP-IB structure, refer to Appendix A of this manual, entitled An HP-IB Overview.

HP-IB Implementation on the 7475

The HP-IB conforms to ANSI/IEEE 488-1978 specifications, and direct interconnection of the HP-IB is via a connector on the rear panel.

The HP-IB functions implemented in the 7475 are as follows:

1. Source Handshake (SH1)
2. Acceptor Handshake (AH1)
3. Talker (T6)
4. Listener (L3)
5. Service Request (SR1)
6. No Remote Local (RL0)
7. Parallel Poll (PP0 if listen-only; PP2 if addr <8; PP1 otherwise)
8. Device Clear (DC1)
9. No Device Trigger (DT0)
10. No Controller (C0)

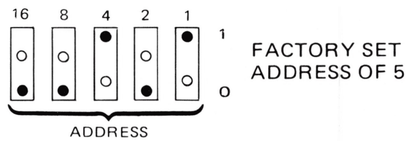
Interface Switches and Controls

The 7475 plotter functions in either of two modes, addressable mode and listen-only mode. In addressable mode, the plotter can function as a talker or as a listener depending on the instructions it receives from the controller. In listen-only mode, it can only listen and it hears all activity on the bus.

Addressing the Plotter

Rear panel switches provide for selection of the plotter address or listen-only mode. Each HP-IB interface can have as many as 15 devices connected to it, set to different specific address codes. The plotter can be set to any one of 31 HP-IB addresses, ranging from 0 through 30. Each address can be selected by setting the switches on the rear panel to the appropriate binary bit positions for the particular address value desired. The address selected establishes the 7475's device address. When using the plotter with an HP desktop computer, do not use 21 which is reserved for the desktop computer's address. When not using an HP desktop computer, be sure the computer and plotter do not have the same address. (Refer to the documentation for your computer.) Address 31 is used to set the plotter to listen-only mode.

The plotter is set to an address code of 05 at the factory. This corresponds to a listen character of % and a talk character of E. Check the following figure for the factory-set address switch positions.



The following table lists the address switch positions for each address value.

Address Characters		Address Switch Settings					Address Codes		
Listen	Talk	16	8	4	2	1	Decimal	Octal	
SP	@	0	0	0	0	0	0	0	
!	A	0	0	0	0	1	1	1	
“	B	0	0	0	1	0	2	2	
#	C	0	0	0	1	1	3	3	
$	D	0	0	1	0	0	4	4	
%	E	0	0	1	0	1	5	5	preset
&	F	0	0	1	1	0	6	6	
’	G	0	0	1	1	1	7	7	
(	H	0	1	0	0	0	8	10	
)	I	0	1	0	0	1	9	11	
*	J	0	1	0	1	0	10	12	
+	K	0	1	0	1	1	11	13	
,	L	0	1	1	0	0	12	14	
−	M	0	1	1	0	1	13	15	
.	N	0	1	1	1	0	14	16	
/	O	0	1	1	1	1	15	17	
0	P	1	0	0	0	0	16	20	
1	Q	1	0	0	0	1	17	21	
2	R	1	0	0	1	0	18	22	
3	S	1	0	0	1	1	19	23	
4	T	1	0	1	0	0	20	24	
5	U	1	0	1	0	1	21	25	Reserved for HP Desktop Computer Address
6	V	1	0	1	1	0	22	26	
7	W	1	0	1	1	1	23	27	
8	X	1	1	0	0	0	24	30	
9	Y	1	1	0	0	1	25	31	
:	Z	1	1	0	1	0	26	32	
;	[	1	1	0	1	1	27	33	
<	\	1	1	1	0	0	28	34	
=	]	1	1	1	0	1	29	35	
>	∧	1	1	1	1	0	30	36	
?	_	1	1	1	1	1	31	37	Sets Listen-only Mode

Bus Commands

Reaction to Bus Commands DCL, SDC, and IFC

The computer can set all devices on the HP-IB system to a predefined or initialized state by sending the device clear command, DCL. The computer can also set selected devices to a predefined or initialized state by sending a selected device clear command, SDC, along with the addresses of the devices. The basic difference is that devices will obey SDC only if they are addressed to listen, whereas DCL clears all devices on the bus. The interface clear command, IFC, is used by the computer to override all bus operations and return the bus to a known quiescent state.

Upon receipt of either a DCL, SDC, or IFC command, the plotter resets the I/O to begin accepting a new instruction, and disables any current output. Any partially parsed HP-GL instruction or parameters will be lost.

The device clear and interface clear commands *do not* reset parameters in the plotter to their default values. They are not the same as the HP-GL instructions, DF or IN.

Serial and Parallel Polling

Polling is the process used by the computer to determine which device on the HP-IB bus has initiated a require service message. The conditions which will cause the require service message to be sent to the computer are defined by the input mask instruction, IM, in Chapter 1.

The Serial Poll

A serial poll enables the computer to learn the status or condition of devices on the bus. It is commonly used by the computer to determine who is requiring service.

The serial poll is so named because the computer polls devices one at a time rather than all at once. The plotter will respond to a serial poll by sending the status byte as described under the output status instruction, OS (Chapter 7). The S-mask parameter of the input mask instruction, IM, is used to specify which status byte conditions will send the service request message and when polled, respond with request service. Unless the user changes the S-mask value from the default setting of 0, the plotter will never give a positive response to a serial poll, i.e., request service (see The Input Mask Instruction, IM, Chapter 1). Bit position 6 of the status byte will be set to 1 (if the S-mask value is not 0) when any of the conditions designated by the S-mask are true. Bit position 6 will be set to 0 after all conditions which would cause a service request no longer exist. See IM, Chapter 1, and OS, Chapter 7. Until bit position 6 has been reset to 0, no additional service request messages, and therefore, no responses to a serial poll are possible.

A computer must issue special commands to initiate and terminate a serial poll. During a serial poll, a device must be instructed to talk and the computer to listen. Therefore, a serial poll cannot be executed when a plotter is in listen-only mode.

The Parallel Poll

Parallel polling can only be done to plotters with an address 0 through 7. Plotters with address settings from 8 through 30 cannot respond to a parallel poll. The plotter will respond positively to a parallel poll only if the conditions specified in the P-mask are satisfied and parallel poll response is enabled. The P-mask parameter of the input mask instruction, IM, is used to specify which status byte conditions will result in a logical 1 response to a parallel poll. The response to a parallel poll is limited to setting the appropriate data line to a logical 1. The line used is determined by the plotter's address value as shown in the table below:

Plotter Address	Parallel Poll Bit Position	HP-IB Data Line Number	
0	7	8	
1	6	7	
2	5	6	
3	4	5	
4	3	4	
5	2	3	Plotter Preset Address
6	1	2	
7	0	1	

To execute a parallel poll, the controller sets the ATN and EOI lines to 1. The controller reads the eight data lines, and determines from these lines which instrument on the bus is requesting service. The computer then sends the parallel poll disable command. Not all computers have parallel poll capability.

It is important to remember that the 7475 will not send a logical 1 unless the P-mask bit value has been changed from the default value of 0 and some condition included in the new P-mask value is true. The plotter does not respond to a parallel poll in listen-only mode.

Positive responses to parallel polls will continue to occur until all bits of the status byte included in the P-mask value have been reset to 0. (See The Output Status Instruction, OS, Chapter 7.)

Addressing the 7475 as a Talker or Listener

To communicate effectively with the 7475 plotter, it is important that you completely understand the addressing protocol of your computer. Therefore, you may wish to review this aspect of your computer before proceeding.

Computers with No High Level I/O Statements

On low level computers, addressing devices on the HP-IB bus is accomplished using mnemonics, such as CMD, which serve as the "bus command."

When bus commands are necessary, a typical addressing sequence is

<Unlisten Command> <Talk Address> <Listen Addresses>

This sequence is made up of three major parts which serve the following purposes:

1. The unlisten command is the universal bus command with a character code of "?". It unaddresses all listeners. After the unlisten command is transmitted, no active listeners remain on the bus.
2. The talk address designates the device that is to talk. A new talk address automatically unaddresses the previous talker.
3. The listen addresses designate one or more devices that are to listen. A listen address adds the designated device as listener along with other addressed listeners.

This basic addressing sequence simply states who is to talk to whom. The unlisten command ("?") plays a vital role in this sequence. It is important that a device receive only the data that is intended for it.

When a new talk address is transmitted in the addressing sequence, the previous talker is unaddressed. Therefore, only the new talker can send data on the bus and there is no need to routinely use an untalk command in the same manner as the unlisten command.

Computers with High Level I/O Statements

In more powerful computers, higher level input/output (I/O) statements are used to specify device addresses on the HP-IB bus. In these cases, the addressing protocol (unlisten, talk, listen) is a function of the computer's internal operating system and need not be of concern to the user.

Sending and Receiving Data

Computer-to-Plotter

Transmitting data from a computer to the plotter is typically accomplished using I/O statements such as WRITE, PRINT, PRINT#, or OUTPUT. The following examples of sending program data to the plotter from various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented by your computer. Each of these examples will cause the plotter to label the identity of the computer sending data, beginning at the X,Y coordinates 1000,2000. The examples involve sending both character string and numeric data as variables, and constants or literals.

HP 9825 and 9826 HPL Example:

```
0: fxd 0;dim A$[13]
1: " SENDING DATA"→A$
2: 2000→Y
3: 9826→B
4: wrt 705,"SP1;PA1000,",Y
5: wtb 705,"LBHP",str(B),A$,3
6: end
```

A terminator is sent by the 9825/9826 at the end of a wrt statement.

Result: HP 9826 SENDING DATA

9826 BASIC Example:

```
10    PRINTER IS 705
20    A$=" SENDING DATA"
30    B=9826
40    Y=2000
50    PRINT "SP1;PA1000,",Y
60    PRINT USING "K";"LBHP ",B,A$,"␃"
70    END
```

A terminator is sent by the 9826 at the end of a PRINT statement.

Result: HP 9826 SENDING DATA

HP 9835/9845 Example:

```
10    PRINTER IS 7,5
20    A$=" SENDING DATA"
30    B=9835
40    C=9845
50    Y=2000
60    PRINT "SP1;PA1000,";Y
70    PRINT USING "K";"LBHP ",B,"/",C,A$,CHR$(3)
80    END
```

A terminator is sent by the computer at the end of a PRINT statement.

Result: HP 9835/9845 SENDING DATA

HP 2647 Example:

```
10 ASSIGN "H#5" TO #1
20 DIM A$[13]
30 A$="SENDING DATA"
40 B=2647
50 Y=2000
60 PRINT #1;"SP1;PA1000,",Y
70 PRINT #1;"LBHP",B,A$,CHR$(3)
80 END
```

A terminator is sent by the 2647 at the end of PRINT #1 statements.

Result: HP 2647 SENDING DATA

HP-83/85 Example:

```
10 PRINTER IS 705
20 A$="SENDING DATA"
30 B=85
40 Y=2000
50 PRINT "SP1;PA1000,",Y
60 PRINT "LBHP";B;A$;"Ñ"
70 END
```

A terminator is sent by the computer following PRINT statements.

Result: HP 85 SENDING DATA

TEK 4051 Example:

```
100 DIM A$[13],B$[1]
110 A$=" SENDING DATA
120 Y=2000
130 B=4051
135 B$=CHR(3)
140 PRINT @5:"SP1;PA1000,";Y;";"
150 PRINT @5:"LBTEK";B;A$;B$
160 END
```

No terminator is sent by the TEK 4051. It must, therefore, be included in each PRINT @ 5 statement if the last HP-GL instruction in the line requires one. In line 140, all characters after the Y may be omitted, since the terminator is optional with the PA instruction.

Result: TEK 4051 SENDING DATA

Commodore PET 2001 and CBM* 8032 Example:*

```
10 OPEN 5,5
20 DIM A$(13)
30 A$=" SENDING DATA"
40 B=2001
50 Y=2000
60 PRINT#5,"SP1;PA1000,";STR$(Y)
70 PRINT#5,"LBPET ";B;A$;CHR$(3)
80 END
```

A terminator is sent by the computer at the end of the PRINT #5 statement.

Result: PET 2001 SENDING DATA

Apple II Applesoft BASIC Example:*

```
10  PR# 3: IN# 3
20 Z$= "WT%" + CHR$ (26)
30  DIM A$(12)
40 A$= " SENDING DATA"
50 Y= 2000
60  PRINT Z$; "SP1;PA1000,",Y
70  PRINT Z$; "LBAPPLE II ";A$;CHR$ (3)
80  PR# 0: IN# 0
90  END
```

*Commodore PET and CBM are trademarks of Commodore Business Machines, Inc. Apple is a trademark of Apple Computer, Inc.

Result: APPLE II SENDING DATA

The PR# 3: IN# 3 statement must be included in each program before instructions can be sent to the plotter. These statements assume the IEEE-488 interface card (HP-IB) is in slot three of the computer. The string Z$ addresses the plotter at address 5 to listen. It must be included in every print statement which sends HP-GL commands to the plotter. The PR# 0: IN# 0 statement directs keyboard output to the display and must be included before the end of the program or before anything can be printed on the display.

Plotter-to-Computer

Outputting data from the plotter to the computer is typically accomplished using I/O statements such as READ, INPUT, or ENTER. Sometimes these statements are only available in I/O ROMs; check your computer's documentation or ask your HP dealer or HP Sales and Support Office. The following examples of obtaining output data from the plotter using various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented on your computer. Each of these examples commands the pen to move to plotter coordinates X = 1000, Y = 1000 and then output the current pen position and the plotter identifier string to the computer.

HP 9825 and 9826 HPL Example:

```
0: fxd 0;dim A$[5]
1: wrt 705,"PA1000,1000;OC"
2: red 705,A,B,C
3: wrt 705,"OI"
4: red 705,A$
5: dsp A,B,C,A$
6: end
```

Displayed current pen position and identification.

1000 1000 0 7475A

HP 9826 BASIC Example:

```
10     PRINTER IS 705
20     PRINT "PA1000,1000;OC"
30     ENTER 705;A,B,C
40     PRINT "OI"
50     ENTER 705;A$
60     DISP A,B,C,A$
70     END
```

Displayed current pen position and identification.

1000 1000 0 7475A

HP 9835/9845 Example:

```
10    PRINTER IS 7,5
20    PRINT "PA1000,1000;OC"
30    ENTER 705;A,B,C
40    PRINT "OI"
50    ENTER 705;A$
60    DISP A,B,C,A$
70    END
```

Displayed current pen position and identification.

1000	1000	0	7475A

HP 2647 Example:

```
10 ASSIGN "H#5" TO #1
20 PRINT #1;"PA1000,1000;OC"
30 READ #1;A,B,C
40 PRINT #1;"OI"
50 READ #1;A$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000	1000	0	7475A

*HP-85/86/87 Example:**

```
10 PRINTER IS 705
20 PRINT "PA1000,1000;OC"
30 ENTER 705 ; A,B,C
40 PRINT "OI;"
50 ENTER 705 ; A$
60 DISP A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000	1000
0	7475A

*Requires I/O ROM, HP Part No. 00087-15003.

TEK 4051 Example:

```
100 DIM A$[5]
110 PRINT @5:"PA1000,1000;OC;"
120 INPUT @5:A,B,C
130 PRINT @5:"OI;"
140 INPUT @5:A$
150 PRINT A,B,C,A$
160 END
```

Displayed current pen position and identification.

1000 1000 0 7475A

Commodore PET 2001 Example:

```
10 OPEN 5,5
20 PRINT#5,"PA1000,1000;OC"
30 INPUT#5,A,B,C
40 PRINT#5,"OI"
50 INPUT#5,A$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000 1000 0 7475A

Commodore CBM 8032 Example:

On the CBM 8032, all alphabetic characters are displayed as lowercase. This is true for both BASIC program statements and for the plotter's response.

A dummy string variable should be included at the end of every input statement which reads data from the plotter because the CBM 8032 sends an untalk command after it receives a carriage return character. Since the plotter with an HP-IB interface terminates all output with a carriage return followed by a line feed, the line feed must be read into this dummy string variable in order to clear the plotter's output buffer for future output.

```
10 OPEN 5,5
20 PRINT#5,"PA1000,1000;OC"
30 INPUT#5,A,B,C,B$
40 PRINT#5,"OI"
50 INPUT#5,A$,B$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000 1000 0 7475a

Apple II Applesoft BASIC Example:

```
10  PR# 3: IN# 3
20 Z$= "WT%" + CHR$ (26)
30 Y$= "RDE" + CHR$ (26)
40  PRINT Z$; "PA1000,1000;OC;"
50  PRINT Y$;
60  INPUT A,B,C
70  PRINT Y$;
80  INPUT D$
90  PRINT Z$; "OI"
100  PRINT Y$;
110  INPUT A$
120  PRINT Y$
130  INPUT D$
140  PR# 0: IN# 0
150  PRINT A,B,C,A$
160  END
```

Displayed current pen position and identification.

1000 1000 0
7475A

For an explanation of PR# 3, Z$ and PR# 0, refer to the Apple II example in the prior section. The string Y$ instructs the plotter at address 5 to talk. The Apple II sends an untalk command after it receives a carriage return character. The plotter with an HP-IB interface terminates all output with a carriage return followed by a line feed. Therefore, in order to clear the plotter's buffer for future output, another talk instruction and another input statement containing a dummy variable (D$ in this program) must follow the input statement which reads parameters of the plotter output statement. The additional talk and input instructions will read the line feed character, thus clearing the plotter's buffer.


CINCH

Chapter 10
RS-232-C/CCITT V.24 Interfacing

What You'll Learn in This Chapter

This chapter is only for 7475 owners with an RS-232-C interface. HP 7475s with Option 001 have an RS-232-C interface.

This chapter describes how to connect the plotter, terminal, and computer in a modem or hardwire environment. It also discusses connecting the interface, pin allocations in the connector, baud rates, stop bits, and transmission errors. It explains four possible operating modes: normal and block modes, and switched lines and leased lines monitoring modes. A tutorial description of the four handshaking methods, hardwire handshake, Xon-Xoff handshake, enquire/ acknowledge handshake, and software checking handshake, is included. The last part of the chapter is devoted to the 14 device control instructions. The syntax of device control instructions is given, followed by a detailed section on each instruction. It is important to be able to use these instructions properly to establish communications with the plotter in your operating environment. You need to master the material in this chapter so you can successfully send HP-GL instructions to the plotter.

NOTE: All information in this chapter applies equally to RS-232-C and CCITT V.24 interfaces. For purposes of simplicity, both are referred to as RS-232-C. ■

Setting Up Your RS-232-C Plotter: a Checklist

The following steps should be followed when interfacing the 7475 plotter with a computer using an RS-232-C interface.

1. Determine which installation and operating environment, described in the first few pages of this chapter, matches your system.
2. Check that you have the required cables and connect the plotter as pictured in the section which describes the environment chosen in step 1. Information necessary when constructing your own cable is found in the section Connecting the RS-232-C Interface.
3. Determine if parity checking is used on your system and set the rear panel parity switches **S1** and **S2** accordingly. Refer to the 7475 Operation and Interconnection Manual.
4. Determine the baud rate at which your computer sends data and set the rear panel switches **B1** through **B4** accordingly. Refer to the 7475 Operation and Interconnection Manual.
5. Determine which handshake your system uses. The four kinds of handshakes are described in the section entitled Handshaking. Note which device control instructions are used to establish that handshake. Since handshaking is often a function of your operating system, you may need to refer to the manuals for your computer to determine which parameters you must set and to what values.
6. In the last part of this chapter, read about the instructions you will use to set up the handshake you have chosen.

Plotter Environments

There are three possible ways to position the 7475 plotter in a computer system. They are described in the following pages; you need only read the section which applies to your system.

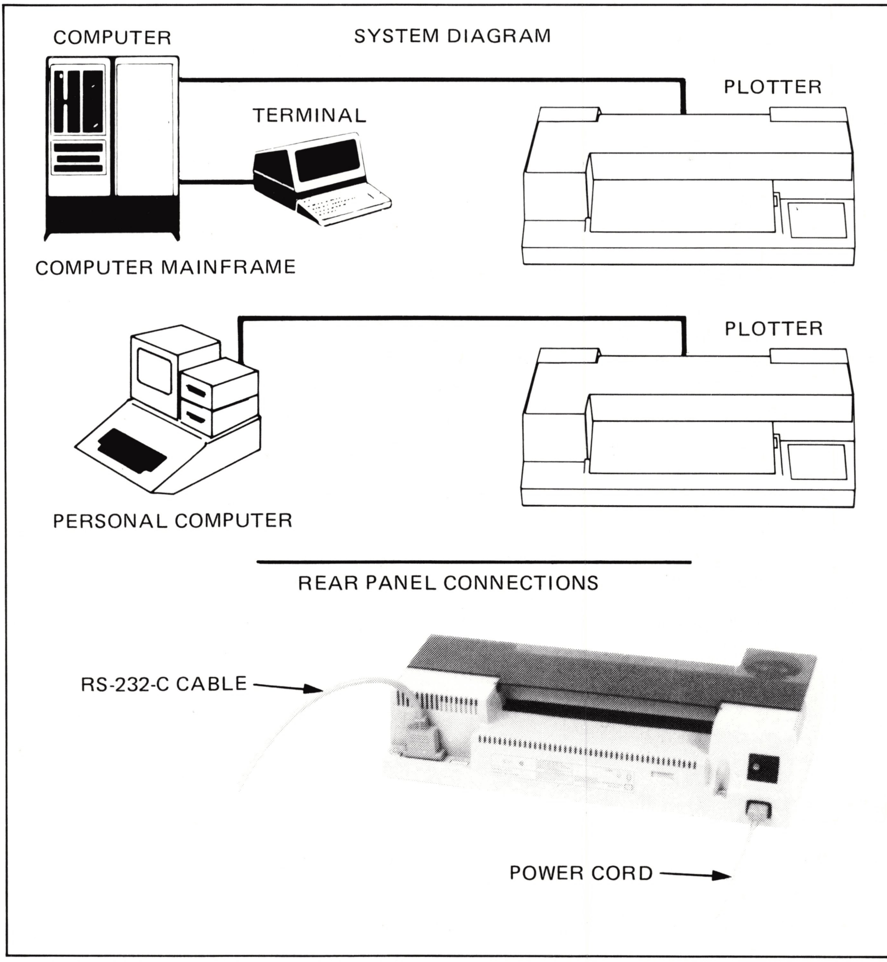
Once the plotter has been connected in a system, it can be placed in an operating state. The operating states which can be accessed in a given environment are described in the operation section for each of the three environments.

Using a Plotter Directly Connected to a Computer Mainframe or Personal Computer

Installation

In this type of system, the plotter is connected directly to a computer and is usually adjacent to it. Entry to the computer is by a keyboard or

terminal through a separate port, rather than through the plotter. This is sometimes referred to as an endline or stand-alone environment. Diagrams of this type of system for both large and personal computers are shown below, along with a picture of the rear panel connection.



Plotter Connection with a Computer Mainframe or Personal Computer

Operation

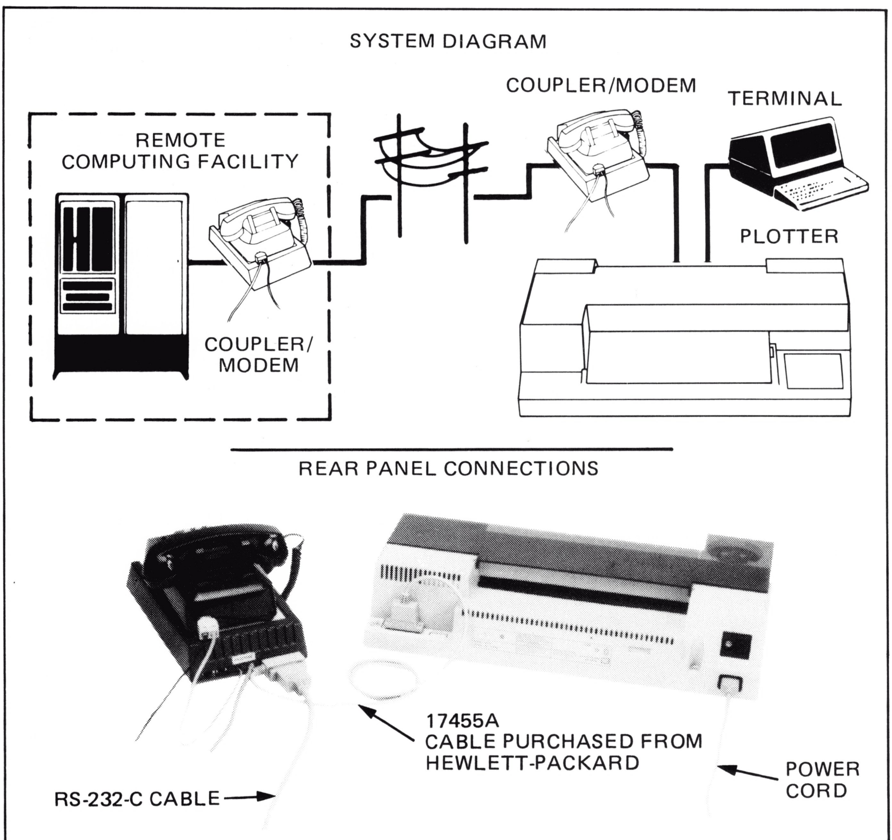
Operation with this type of installation is usually confined to the on-line, programmed-on state. The rear panel switch labeled **Y/D** should be set to **D** (direct). When the switch is set to **D**, whenever power is being applied to the plotter, it is in the on-line, programmed-on state. In this state, the plotter reacts to all device control and HP-GL instructions except the plotter off instruction. It is not possible to programmatically turn the plotter off. Only when the switch is set to **Y** may the plotter be

placed in the on-line, programmed-off mode. That operating state is described under operation with a terminal.

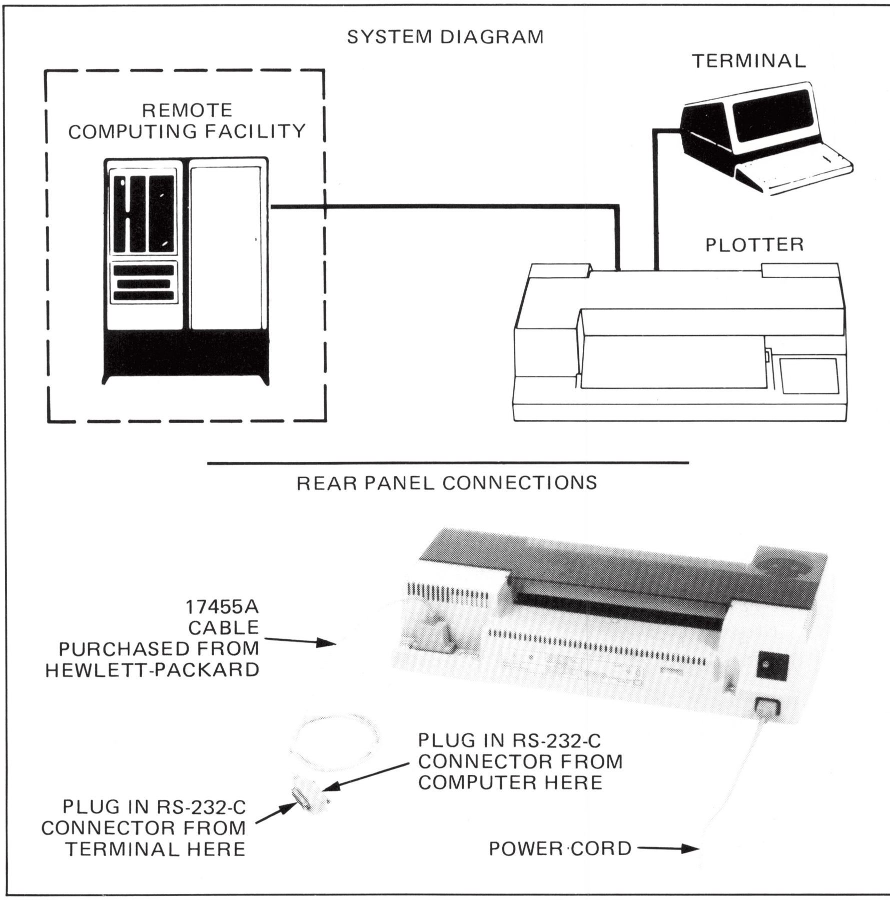
Using a Plotter in an Environment with a Terminal

Installation

In the second type of system, the plotter is connected in series between the computer and the terminal. The plotter's **LINE** switch must be **ON** in order to have any communication between the terminal and the computer. There may be a direct wire between the computer and the plotter or the plotter may be connected to a modem and communication may take place over telephone lines. This setup, with the plotter between the computer and the terminal, is sometimes referred to as eavesdrop environment. A special Y-cable (Part No. 17455A), which joins the lines from the computer and terminal into the plotter's one connector, must be used in this environment. Diagrams of the two systems, with and without a modem, follow, along with pictures of the rear-panel connections for both kinds of systems.



Plotter Interconnection with a Terminal and Remote Facility Using Modems



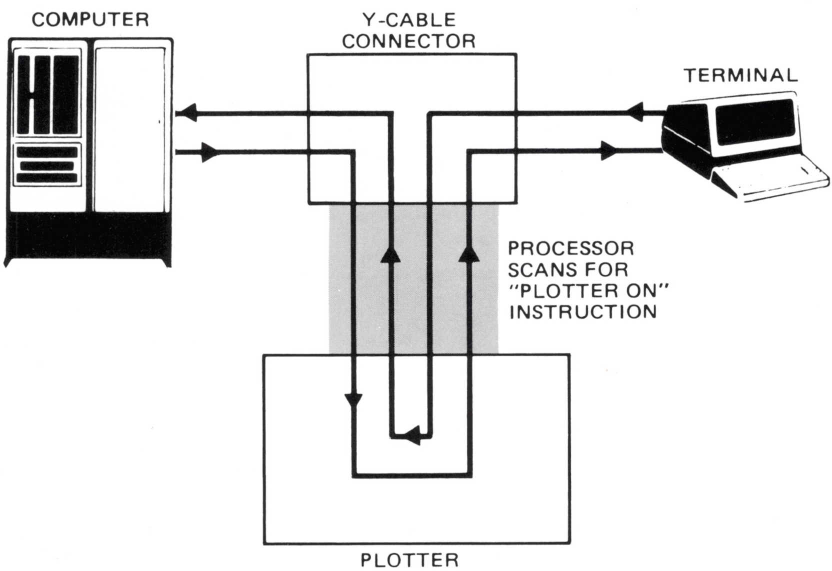
Plotter Interconnection with a Terminal and Remote Facility Using RS-232-C/CCITT V.24 Cabling

Operation

While operating in this environment, the plotter may be in one of three states: on-line, programmed-off; on-line, programmed-on; or monitor mode.

On-Line, Programmed-Off State

The plotter can only be in this state if the Y/D switch on the rear panel is set to Y (used with Y-cable). When this switch is set to Y, the plotter is placed in the on-line, programmed-off state by either turning the plotter's LINE switch to ON or by receipt of a plotter off instruction from the computer or of a terminal-generated Break signal while the plotter is in the on-line, programmed-on state. In the on-line, programmed-off state, the plotter's processor passes data between the computer and the terminal as shown in the following diagram. The plotter will respond only to a plotter on instruction from the host computer.



Plotter in On-Line, Programmed-Off State

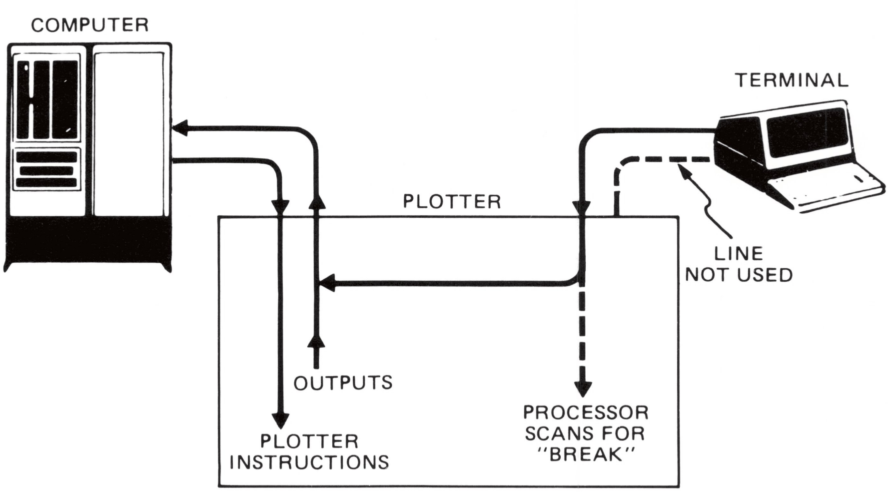
On-Line, Programmed-On State

When the rear-panel switch labeled Y/D is set to D, the plotter is placed in the on-line, programmed-on state by turning on the plotter. When the Y/D switch is set to Y, the plotter is switched from the on-line, programmed-off state to the on-line, programmed-on state when a plotter on instruction, ESC . (or ESC . Y, is received from the computer.

When in this state, the plotter operates in response to instructions received from the computer as shown in the following figure. When the plotter instructions request output, it is provided as shown. The communication channel from the terminal to the computer, through the plotter, is maintained to provide operator entry into the computer.

The plotter's processor monitors the channel from the terminal to the computer for a terminal-generated Break signal. The plotter will interpret anything greater than a 130-millisecond space as a Break. This Break signal is retransmitted to the computer and in-process plotter outputs are aborted, but plotting continues until stored buffer data is completed. A new plotter on instruction from the computer is required to resume plotting operations. The plotter will ignore a Break signal if the Y/D switch is set to D.

It should be noted that in the on-line, programmed-on state (but not in monitor mode which is described in the next paragraph) all data generated by the terminal are routed through to the computer on a noninterference basis when the plotter is not doing outputs. Data generated by the terminal are ignored while output is occurring. However, all data generated by the computer are intercepted by the plotter and not passed to the terminal.

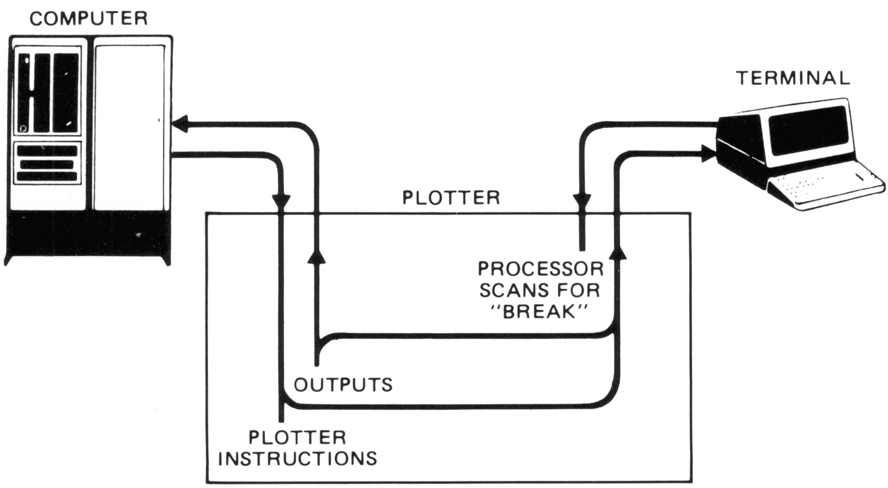


Plotter in On-Line, Programmed-On State

Monitor Mode

After the plotter is in the on-line, programmed-on state, two mutually exclusive monitor modes may be enabled using the set plotter configuration instruction, ESC . @. Depending upon which monitor mode is enabled, either all data (including device control instructions) are retransmitted to the terminal CRT or only HP-GL data are retransmitted as they are parsed from the plotter's buffer. All plotter output responses are sent to both the computer and terminal. Refer to The Set Plotter Configuration Instruction, ESC . @, for complete information.

The plotter monitors for a terminal-generated Break signal. Receipt of a Break signal will cause the same results as described under the on-line, programmed-on state. Then, new plotter on and set plotter configuration instructions from the computer are required to resume plotting operations with monitor mode active. The following diagram shows how the plotter processes data while in monitor mode.

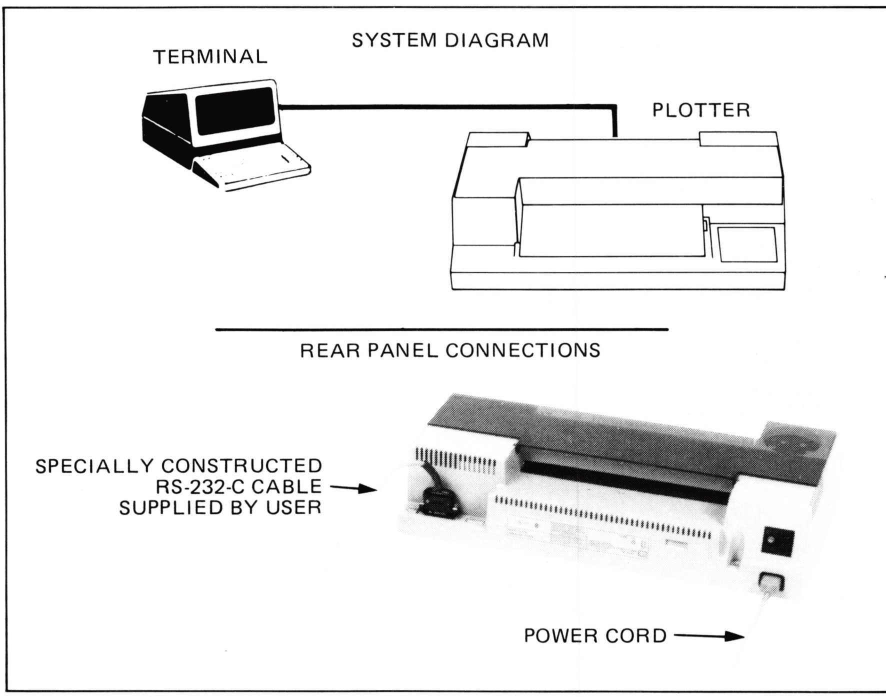


Monitor Mode

Using the Plotter in a Terminal-only Environment

Installation

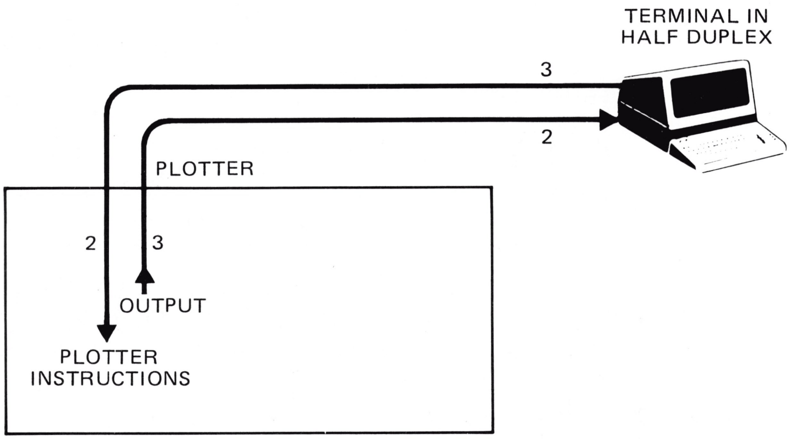
The 7475 plotter can be directly connected to a terminal if a specially constructed, user-supplied cable is used that cross connects lines 2 and 3. While there is no computer in this configuration, the terminal usually has some "intelligence." When the terminal and plotter are connected using this special cable, the terminal may be used to send instructions to the plotter. A diagram of the terminal-only environment and a picture showing the rear-panel connection follow.



Plotter Interconnection with Only a Terminal

Operation

The rear-panel switch labeled **Y/D** should be set to **D**. If it is set to **Y**, the plotter must receive a plotter on instruction, ESC **.** (or ESC **.** Y, before it will respond to other instructions from the terminal. The terminal should be set to half duplex in order to view the characters being sent to the plotter. Plotter output will be displayed on the terminal. The following diagram shows plotter operation when in the programmed-on state in a terminal-only environment.



Terminal-only Environment, Programmed On

Connecting the RS-232-C Interface

The 7475 plotter interfaces to the RS-232-C communications lines through a standard 25-pin female connector mounted on the back of the plotter. The 7475 is capable of operating in a three-wire (transmit, receive, ground) configuration.

In hardwired handshake operation, the Data Terminal Ready line (pin 20 of the connector on the plotter) is used to monitor the space in the buffer available for input. The plotter outputs data when requested (refer to Hardwire Handshake in this chapter).

If you are fabricating the cable assembly, the connector should be a 25-pin type "D" subminiature CINCH DBC-25P plug or equivalent.

Connector pin allocations for the three-wire configuration are identified and described in the following table.

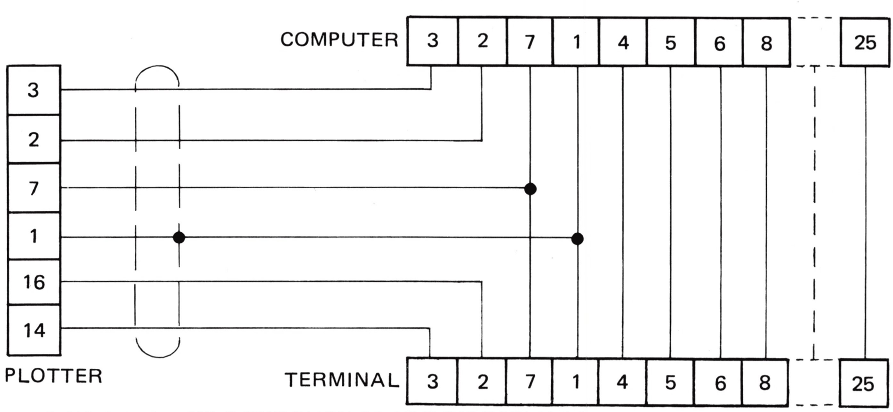
Minimum Interface Connector Pin Allocations

Pin No.	RS-232-C	CCITT V.24	Function/Signal Level
2	BA	103	Transmitted Data High = SPACE = "0" = +12 V Low = MARK = "1" = −12 V
3	BB	104	Received Data High = SPACE = "0" = +3 V to +25 V Low = MARK = "1" = −3 V to −25 V
7	AB	102	Signal ground (Common return)

In addition to the minimum requirements for communication, ten more lines are connected as shown in the following table. These lines are required to implement full duplex communication, intermediate baud rate, hardwired handshake mode, and monitor mode. All remaining pins make no internal connection.

Pins 14 and 16 are wired in the special Y-cable, available as Option 16, to implement monitor mode. The Y-cable schematic is shown below.

NOTE: Hardwire handshake **cannot** be used to prevent buffer overflow when the Y-cable is connected. This is because pin 20 is connected between the **COMPUTER** and **TERMINAL** connectors, but not to the **PLOTTER** connector. ■



PINS 4, 5, 6, AND 8 THROUGH 25 ARE DIRECTLY CONNECTED BETWEEN THE **COMPUTER** AND **TERMINAL** CONNECTORS.

Y-cable Schematic

Additional Connector Pin Allocations

Pin No.	RS-232-C	CCITT V.24	Function/Signal Level
1	AA	101	Protective ground
4	CA	105	Request To Send from the plotter Always High = ON = +12 V
5	CB	106	Clear to Send High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
6	CC	107	Data Set Ready High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
8	CF	109	Received Line Signal Detector High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
17	DD	115	External Clock Input High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
20	CD	108.2	Data Terminal Ready to modem High = ON = +12 V Low = OFF = −12 V
23	CH/CI		Data Signal Rate Selector Always High = ON = +12 V
14*	SBA	118	Secondary Transmit Data Data line from plotter to terminal
16*	SBB	119	Secondary Received Data Data line to plotter from terminal

*Used to establish monitor mode with special Y-cable (Part No. 17455A).

For FTZ/European applications, two additional modes are available: switched lines monitoring mode and the leased lines monitoring mode.

In the switched lines monitoring mode, the CC and CB will be monitored. If either of these lines go low, the CD line will be driven low by the plotter to automatically disconnect the channel from the line. This mode is enabled by depressing the pen 5 pushbutton on the front panel during power-up.

In the leased lines monitoring mode, the CC, CB, and CF will be monitored. If any of these lines go low, the CD line will be driven low by the plotter to automatically disconnect the channel from the line. This mode is enabled by depressing the pen 6 pushbutton on the front panel during power-up.

If none of the pen pushbuttons are depressed at power-up, the plotter is in normal mode (3-wire mode).

NOTE: If you are using an eavesdrop cable and you set up a switched line monitoring mode or a leased lines monitoring mode, the plotter will not be able to monitor the other signal lines such as CB, CC, CF, and DTR, and you will not be able to output data. Also, if either switched lines monitoring mode or leased lines monitoring are operational, you cannot use hardwire handshake. ■

Output Baud Rate

The plotter is designed to operate in an asynchronous mode with switch-selectable baud rates of 75, 110, 150, 200, 300, 600, 1200, 2400, 4800, and 9600. However, setting all **BAUD** switches to zero and connecting an external clock input to pin 17 of the connector allows operation of the plotter at any intermediate baud rate up to 9600 baud. Both the receiver (RRC) and transmitter (TRC) clocks will operate at the same clock rate. Requirements for the clock signal are as follows:

1. The clock frequency must be 16 times the desired baud rate.
2. The baud rate must not exceed 9600.
3. The duty cycle of the clock pulse should be close to 50%.
4. The clock pulse must be a logic on of $+2\text{ V} < \text{V} < 25\text{ V}$ and a logic off of $-25\text{ V} < \text{V} < +0.8\text{ V}$ ($3.5\text{ k}\Omega$ input impedance).
5. Care should be taken to keep the transmission lines as short as possible to minimize transmission line reflection noise.

Stop Bits

The plotter is configured to automatically verify or generate one or two stop bits, depending on the setting of the plotter's baud rate switches. Refer to the 7475A Operation and Interconnection Manual for more information.

Transmission Errors

Transmission errors occur when communication between the computer and plotter is incomplete or does not conform to what is expected or required by either party.

Transmission errors include:

- Framing error — the plotter does not detect a valid stop bit at the end of every character.
- Parity error — the plotter does not detect the expected parity (odd or even).
- Overrun error — a plotter instruction writes over another instruction.
- Buffer overflow error — the plotter receives more bytes of data than it has space for in the buffer.

When the plotter detects a framing, parity, or overrun error, it turns on the front panel **ERROR** light and sets error code 15. This error code generally indicates that the communication incompatibility is hardware related (incorrect stop bit jumper installation, wrong parity selection, incompatible or incorrectly set baud rates, etc.).

When the plotter detects a buffer overflow, it turns on the front panel **ERROR** light and sets error code 16. The last HP-GL data that caused the overflow will be lost. Error code 16 generally indicates an improperly established handshake protocol.

The **ERROR** light remains on until either the user interrogates the plotter via an output extended error instruction, ESC . E, and the plotter responds with the appropriate error code, or the user turns the plotter off, or an HP-GL initialization instruction, IN, is processed, or a front-panel reset occurs.

A complete list of error codes is included with the discussion of the ESC . E instruction.

NOTE: A buffer overflow condition may also cause an HP-GL error to occur. In this case, an HP-GL IN or OE instruction or a front-panel reset must be executed in order to clear the **ERROR** light. See Chapter 7 for an explanation of the output error instruction, OE. ■

Handshaking

The 7475 uses a 1024-byte input buffer to synchronize the processing of data with the rate at which it is received. The presence of an input buffer requires that the computer and the plotter transfer information to one another in such a way that data will not be lost or misinterpreted. This is the purpose of handshaking.

The 7475 is capable of using any one of four handshaking methods to prevent buffer overflow and the resulting loss of data. The computer system's capabilities and requirements dictate which handshake method is appropriate.

- Hardwire Handshake — uses a physical wire, pin 20 of the RS-232-C cable, to control handshaking. It can be used if the computer system can or does monitor pin 20 (DTR).
- Xon-Xoff Handshake — is managed by the peripheral device. It can be used if the computer system follows an Xon-Xoff protocol (control characters are transmitted from the peripheral to the computer).
- Enquire/Acknowledge Handshake — is managed by the computer system and interface. This handshake is often used in Hewlett-Packard systems and is so named because the ASCII characters ENQ and ACK may be used to control the handshake.
- Software Checking Handshake — is managed by the applications programmer. It can be used on almost any computer system, but it must be used if the system cannot implement any of the other three handshaking methods.

Once the handshake method is selected, the 7475 can be programmatically instructed to match the computer system requirements, implement the chosen handshake method, and function properly within the system-dependent communication environment. This is done by specifying certain variables in device control instructions which are issued to the 7475 at the beginning of each computer session or graphics program. The variables, which may be specified by using the decimal value of the character desired to establish one of the four handshake methods available to the 7475, are:

- Output Trigger Character — The output trigger character, when used, is the last character output by the computer when making a request of a graphics peripheral. Defining this character in an instruction tells the plotter, "Don't respond to my request until you receive this trigger character." This character is often a DC1 (decimal equivalent 17) or some other nonprinting ASCII character such as LF or CR or, when using some implementations of BASIC, the ? (decimal equivalent 63), which does print.

- Turnaround Delay — The turnaround delay is the length of time the plotter will wait after receiving a computer request and the trigger character, if any, before it responds. The purpose of this time delay is to delay the plotter's transmission of requested data until the computer is ready to receive and process it. Systems may require either a turnaround delay or a trigger character, or both.

- Output Initiator Character — The output initiator character is a one-character initiator that is sent by the plotter at the beginning of a string. The output initiator tells the computer, "This starts my transmission." Some computers which require an output initator expect the start-of-text character, STX (decimal equivalent 2), as the plotter's output initiator.

- Output Terminator — The output terminator is a one- or two-character terminator that the computer requires the plotter to send at the end of each response to a data request. The output terminator tells the computer, "This completes my transmission." Often, computers expect the carriage return character, CR (decimal equivalent 13), as the plotter's output terminator.

- Echo Terminate Character — Echoing is commonly found in full duplex systems. Use of the echo terminate parameter in a device control instruction tells the plotter that the computer will echo all responses and that this echoed data should be ignored (the plotter's data buffer should be closed) until an echo terminate character is received. When the plotter receives the echo terminate character, it reopens the data buffer to receive graphics data from the computer. Computers often use the line feed character, LF (decimal equivalent 10), as the echo terminator. If the computer does not echo the peripheral's response, this variable must be zero (equivalent to null) or must be omitted.

- Intercharacter Delay — Some computers cannot process data as fast as the plotter can transmit it due to limited buffering in the I/O port. This can be compensated for by delaying each transmission from the plotter a period of time as specified by the intercharacter delay variable. This intercharacter delay is added to a turnaround delay (if one has been specified) before the first character is sent by the plotter, and is also inserted before each subsequent character in a string being sent to the computer.

- Enquiry Character — In some systems the computer sends an enquiry character to ask the plotter if it has room for a block of data, thereby initiating the handshake process. If Xon-Xoff handshake mode is to

be established, a NULL character (decimal equivalent 0) must be specified as the enquiry character. If enquire/acknowledge is to be established, an ENQ character (decimal equivalent 5) or any other ASCII character besides the NULL is used.

- Immediate Response String — Certain system environments require an immediate response from the plotter acknowledging the enquiry from the computer. Systems of this type include a computer that transmits data to the plotter after a certain time interval but before receiving a go-ahead signal from the plotter. If the plotter's buffer is full and the computer sends more data, the buffer will overflow. The immediate response string prevents this inadvertent transmission of data before the plotter is ready. It is transmitted by the plotter immediately after receipt of an enquiry character and tells the computer, "Wait, I am here and checking my buffer space." Computers frequently require a DC3 character (decimal equivalent 19) for the immediate response.

- Acknowledgment String — The acknowledgment string specifies the character or characters that the plotter will send to the computer when the plotter's input buffer has room for another block of data. Computers frequently require that the ACK character (decimal equivalent 6) be used for the acknowledgment string.

- Data Block Size — This is the maximum size of each data block the computer will transmit to the plotter.

- Data Terminal Ready (CD) Line Control — This variable sets the configuration of the plotter's Data Terminal Ready control line (pin 20) to enable or disable the hardwire handshake mode. Pin 20 is held on (+12 V) if hardwire handshake is disabled.

- Xoff Threshold Level — In the Xon-Xoff handshake mode this defines how many empty bytes remain in the buffer when the plotter sends the Xoff trigger character to the computer, telling it to stop sending data.

- Xoff Trigger Character — This specifies the character string the plotter will use to signal the computer to temporarily stop sending data while the plotter processes what it has already received. The DC3 character (decimal equivalent 19) is generally used for the Xoff trigger.

- Xon Trigger Character — This specifies the character string the plotter will use to signal the computer that there is sufficient space in the buffer to resume sending data. The DC1 character (decimal equivalent 17) is generally used for the Xon trigger.

The following discussion of the four handshake methods includes the pertinent variables and identifies the instructions which will establish their values.

Software Checking

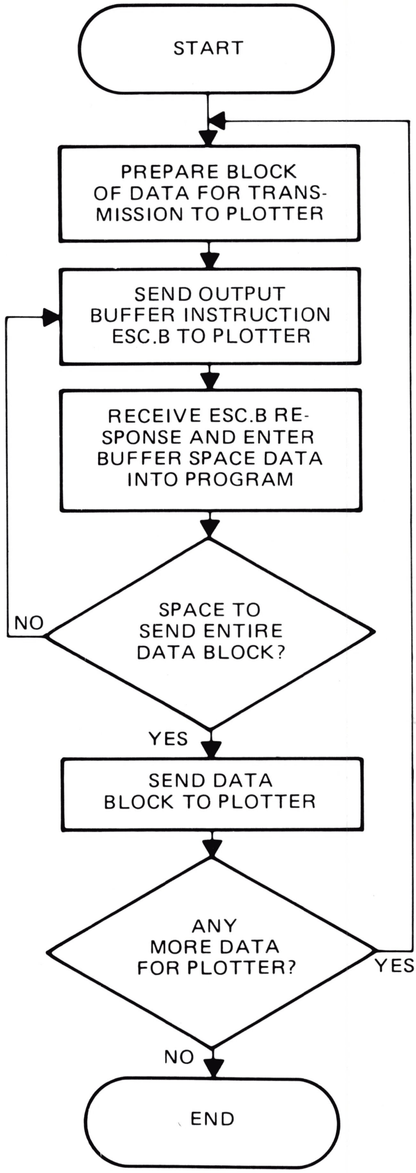
Software checking is a nonautomatic handshake method in which the user's program repeatedly asks the plotter how many characters of empty space remain in the buffer. When the plotter response is bigger than the next block of data, the program will transmit the data block to the plotter. This method is inefficient in time-share environments.

The advantage of software checking is that it is independent of hardware and operating system abilities required to implement other handshake modes; therefore, it usually makes software transportable between computer systems. The limitation of this method of handshaking is that it uses up computer time.

To match the requirements of the computer system, these variables may be specified for the software checking handshake mode by using the appropriate instruction:

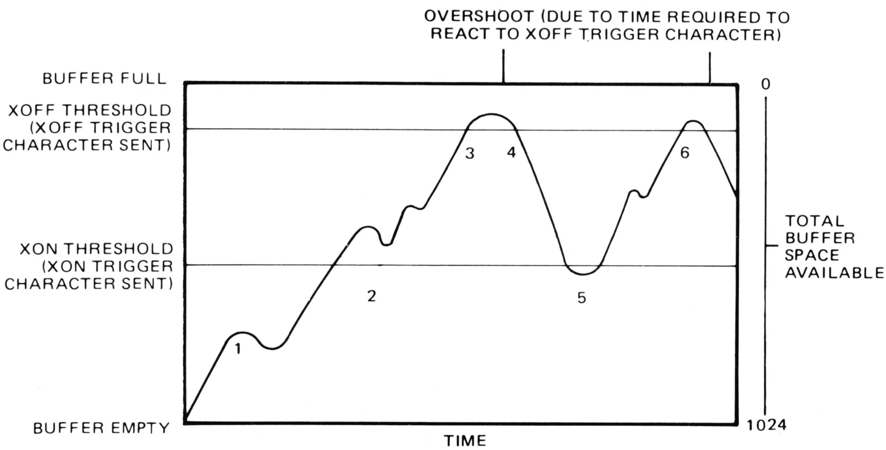
- Turnaround delay (ESC . M instruction)
- Output trigger character (ESC . M instruction)
- Echo terminate character (ESC . M instruction)
- Output initiator character (ESC . M instruction)
- Output terminator (ESC . M instruction)
- Intercharacter delay (ESC . N instruction)

The following flow diagram illustrates the functional elements of a typical software checking handshake within a user's program.



Xon-Xoff Handshake

With the Xon-Xoff handshake method, the plotter controls the data exchange sequence by telling the computer when it has room in its buffer for data and when to shut off the flow. The plotter uses buffer threshold indicators (an Xon trigger character and an Xoff trigger character) to prevent buffer overflow.



Xon-Xoff Threshold Levels

As data is sent to the plotter by the computer, it is stored in the buffer and simultaneously acted on by the plotter. The preceding figure is representative of the way the Xon-Xoff handshake works; the numbers represent the following:

1. Data enters the buffer faster than it can be acted on by the plotter, and the buffer starts to fill.
2. The plotter begins processing the input data faster than the computer sends it, and the buffer starts to empty.
3. The data enters the buffer at a faster rate than the plotter can process it. The amount of data stored in the buffer reaches the Xoff threshold level, at which point the plotter sends the Xoff trigger character stopping the flow of data from the computer.
4. Due to a finite delay between the time the plotter sends the Xoff trigger character and the time it takes the computer to react, a slight overshoot may occur. For this reason, the Xoff threshold level should always be specified at least as large as the data block size or the

maximum number of bytes sent by an output statement to allow room for the overshoot.

5. Once the Xoff trigger character has been sent, when the amount of stored data drops to the Xon threshold level, the plotter sends the Xon trigger character to signal the computer to resume sending data. The Xon threshold level is automatically set at 512 bytes. If the Xoff threshold level is greater than 512, the Xon threshold is reset to send the Xon character when one more byte than required by the Xoff threshold is available in the plotter's buffer.
6. Data is again stored in the buffer until all the data are transferred or until the Xoff threshold level is exceeded again.

The following conditions can be specified for the Xon-Xoff handshake mode to match the requirements of the computer system, by using the appropriate instruction:

- Xoff threshold level (ESC . I instruction)
- Xon trigger character (ESC . I instruction)
- Xoff trigger character (ESC . N instruction)
- Intercharacter delay (ESC . N instruction)

The enquiry character (ESC . I instruction) must either be defaulted or specified as zero.

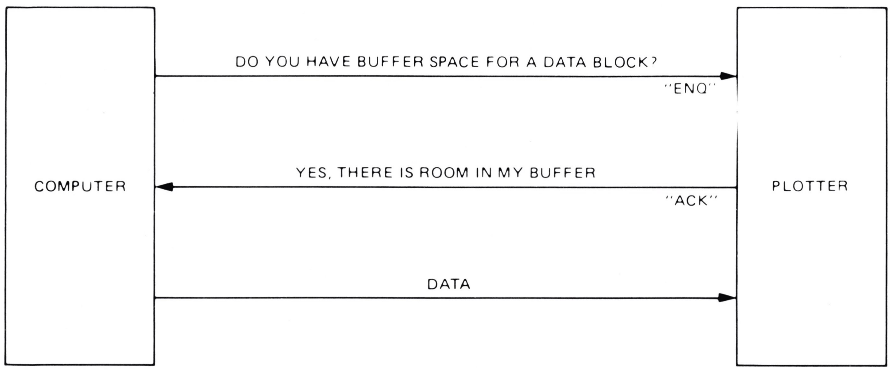
Enquire/Acknowledge Handshake

With the enquire/acknowledge handshake, the computer's operating system or application program initiates the data exchange process by querying the plotter about the availability of buffer space. The format of the exchange is dependent upon the requirements of the computer. The following conditions can be specified for the enquire/acknowledge handshake mode by using the appropriate instruction:

- Turnaround delay (ESC . M instruction)
- Output trigger character (ESC . M instruction)
- Echo terminate character (ESC . M instruction)
- Output initiator character (ESC . M instruction)
- Output terminator (ESC . M instruction)
- Intercharacter delay (ESC . N instruction)
- Immediate response string (ESC . N instruction)
- Data block size (ESC . I or ESC . H instruction)

- Enquiry character (ESC . I or ESC . H instruction)
- Acknowledgment string (ESC . I or ESC . H instruction)

In its simplest form, the data exchange looks like this:



ENQ/ACK Handshake Protocol Example 1

In a more complex form, the communication might look like the following example, where the two instructions ESC . M250;17;10;13: and ESC . H100;5;6: have been sent to specify the variables as:

turnaround delay = 250 ms

output trigger character = ASCII character DC1 (decimal equivalent 17)

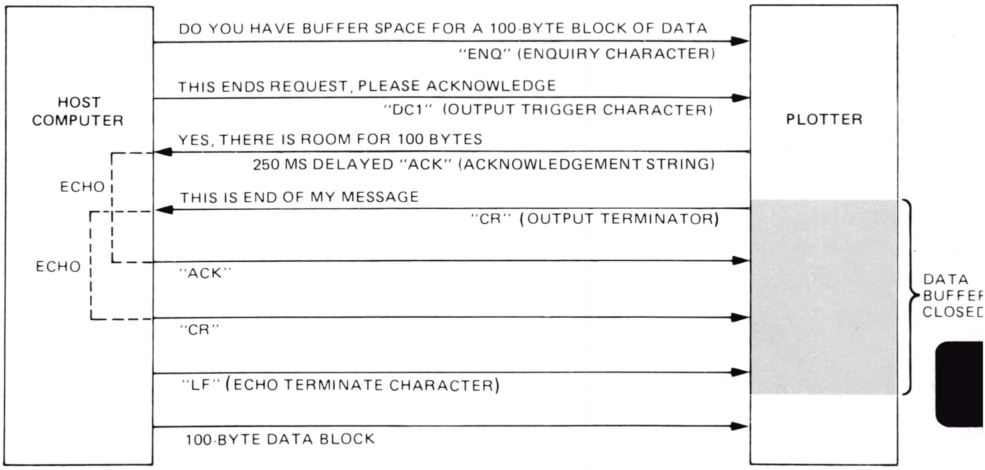
echo terminate character = ASCII character LF (decimal equivalent 10)

output terminator = ASCII character CR (decimal equivalent 13)

data block size = 100 bytes

enquiry character = ASCII character ENQ (decimal equivalent 5)

acknowledgment string = ASCII character ACK (decimal equivalent 6)



ENQ/ACK Handshake Protocol Example 2

Hardwire Handshake

As the name implies, the hardwire handshake takes place in the hardware rather than the firmware or software. The plotter controls the data exchange sequence by setting the electrical voltage on pin 20 of the connector (CD line) to the computer to signal the computer when to send another block of data. If there is enough room in the plotter's buffer to accept and store another block of data, the plotter sets the Data Terminal Ready, CD, line to a high state. If there is insufficient space, it sets the line low. By monitoring this line, the computer knows when it can or cannot safely transmit another block of data.

The hardwire handshake mode is enabled at power on or by setting the Data Terminal Ready, CD, line control using the ESC . @ instruction.

Data Transmission Modes

The RS-232-C version of the 7475 has two modes of data transmission: normal mode and block mode.

Normal Mode

In normal mode, all HP-GL instructions are put in an execution buffer where they are parsed and executed in order. Escape sequence instructions (ESC . E, ESC . B, ESC . M, etc.) are not buffered but are executed immediately.

Block Mode

In block mode, all characters received are put in an intermediate buffer except for escape sequence instructions and handshake characters. Escape sequence instructions are still executed immediately.

The size of a block is variable and is not defined explicitly. A new block is started after an ESC . E instruction has been received. A block of instructions is terminated by the receipt of another ESC . E instruction which outputs the current RS-232-C error state. Refer to the ESC . E instruction for additional information.

Block mode has no effect on the type of handshaking used or on the handshaking parameters that are defined. The number of characters in the intermediate buffer plus the number of characters in the execution buffer cannot exceed a total of 1024 characters.

Access block mode by setting bit 4 in the second parameter of the ESC . @ instruction. Set this bit to 0 for normal mode and to 1 for block mode. Refer to the discussion of the ESC . @ instruction for additional information. Use block mode to catch transmission errors that have reached the plotter. This allows you to retransmit the correct block of data, and thus prevent errors to the plot.

When the 7475 powers up, it is in normal mode. When block mode is turned on, any characters in the buffer are put in the execution buffer. When block mode is turned off, the state of the intermediate buffer is undefined. Before turning off block mode, send an ESC . E instruction to clear out the intermediate buffer.

NOTE: It is not advisable to use the ESC . L instruction when in block mode as it may disrupt communication. If you use an ESC . L instruction while in block mode, use it immediatley after an ESC . E instruction. Send the ESC . E instruction, read the response, send the ESC . L instruction, read the response, and then send the additional HP-GL instructions. ■

RS-232-C Device Control Instructions

Device control instructions establish the handshake protocol to be used by the 7475 plotter. All communications conform to the protocol established by these instructions. The instructions serve two purposes: to control the method by which data is transferred between the computer and the plotter (input/output operations), and to give the computer the ability to query and to receive information from the plotter.

Each instruction's name gives an immediate clue to its purpose: if "output" is the first word in the name of the instruction, the computer wants a response from the plotter. Otherwise, the instruction concerns the I/O functions. The word "set" in the title indicates the instruction conditions under which subsequent I/O is to occur.

The plotter acts on device control instructions immediately upon receipt. It does not store them in the data buffer.

Syntax for Device Control Instructions

Device control instructions are three-character escape code sequences comprised of "ESC" and "." followed by one of the characters @, B, E, H, I, J, K, L, M, N, or O, R,), (, Y, or Z.

These syntax conventions are used with the instructions discussed in this chapter:

[ ]	Brackets indicate that all parameters enclosed are optional.
( )	Parentheses indicate that each individual parameter is optional.
;	The semicolon follows and delimits parameters. If a semicolon appears without a parameter, the parameter is defaulted.
:	The colon terminates any instruction which may have parameters and can occur after any valid number of parameter entries. Any parameter that is not specified is defaulted.
<DEC>	This symbol specifies a decimal value parameter. For example, the characters 10 would represent the decimal value ten; the characters 13 would represent the decimal value thirteen.
<ASC>	This symbol specifies the decimal equivalent for an ASCII character (see the ASCII Character Equivalents table in Appendix C). In this case, the characters 10 would represent the ASCII line feed character, LF, and 13 would represent the ASCII carriage return character, CR.
. . .	Specifies a number of optional parameters. Each parameter must be followed by a delimiter (;) or the terminator (:).
TERM	Unless changed by an ESC . M instruction, all RS-232-C output responses include a CR as a terminator.

Default Values; Omitting Parameters — Any parameter may be omitted or, if the parameter is required, it can be set to its default value by omitting the parameter and entering only the semicolon as a delimiter. All parameters may be omitted and therefore set to default values by entering only the colon terminator after the instruction.

ESC — Denotes the single ASCII character, Escape, which in most computers is accessed by striking a single key on the keyboard.

NOTE: There is no delimiter (semicolon) between the three-character command sequence, e.g., ESC . O, and the first parameter. ■

The Plotter On Instruction, ESC . (or ESC . Y

DESCRIPTION The plotter on instruction, ESC . (or ESC . Y, places a plotter which is powered on into the on-line, programmed-on mode so that it will accept incoming data and interpret it as plotter instructions.

USES This instruction is used when the rear-panel switch labeled Y/D is set to Y to ready the plotter to accept other instructions. It is sent at the beginning of any plotting program or when the user wishes to resume plotting after the plotter has been turned off by an ESC .) or ESC . Z instruction or a Break.

SYNTAX ESC . (

or

ESC . Y

EXPLANATION This instruction is ignored when the rear-panel switch labeled Y/D is set to D since, in that case, turning on the power places the plotter in the programmed-on state.

Beginning with the next character, the plotter will accept incoming data and interpret it as plotter instructions. If the plotter is already in the programmed-on state, it will ignore this instruction.

The Plotter Off Instruction, ESC .) or ESC . Z

DESCRIPTION The plotter off instruction, ESC .) or ESC . Z, takes the plotter out of on-line, programmed-on state so that it neither accepts nor interprets incoming data until another plotter on instruction is received.

USES The instruction is used to deactivate the plotter. It is used at the end of a graphics program or in some environments to allow data to be passed through the plotter to the terminal.

SYNTAX ESC .)

or

ESC . Z

EXPLANATION This instruction is ignored when the rear-panel switch labeled Y/D is set to D. When that switch is set to D, it is not possible to turn the plotter off programmatically.

Beginning with the next character, the plotter will assume a passive state and remain in that state until a plotter on instruction is received.

Any HP-GL instructions remaining in the buffer at the time that a plotter off instruction is received are executed. However, no additional HP-GL instructions will be accepted by the plotter.

NOTE: A Break signal from the terminal will have the same effect as a plotter off instruction. ■

The Set Plotter Configuration Instruction, ESC . @

DESCRIPTION The set plotter configuration instruction, ESC . @, specifies an effective maximum buffer size, and sets parameters necessary for hardwire handshake mode, monitor mode, and Data Transmission Mode.

USES The instruction is used to set up an effective maximum buffer size, to enable or disable hardwire handshake or monitor mode, and to establish Data Transmission Mode.

SYNTAX ESC . @ [(<DEC>) ; (<DEC>)] :

DEFAULT ESC . @ : Sets up default buffer size (1024 characters), enables hardwire handshake, disables monitor mode, and leaves the Data Transmission Mode unchanged.

EXPLANATION A description of the instruction's parameters follows:

<DEC> The first parameter is not required; if a parameter is included, it specifies an effective maximum buffer size. Parameter range is 0 to 9999. A parameter equal to or greater than 1024 is interpreted as 1024. The semicolon must precede any second parameter.

<DEC> Only bits 0, 2, 3, and 4 are used. Bit 0 of the second parameter establishes hardwire handshake with Data Terminal Ready, CD, line control. Bit 2 establishes the

type of monitor mode. Bit 3 set to 0 disables monitor mode; set to 1 enables the monitor mode established by bit 2. Block mode is enabled by setting bit 4 in the second parameter to 1. Setting bit 4 to 0 enables normal mode. Refer to the discussion of block mode in this chapter for additional information. If the second parameter is not specified, the Data Transmission Mode is unchanged.

The following chart describes the second parameter bit functions.

Bit No.	Logic State	Description
0	0	Set and hold line high (disable hardwire handshake).
	1	Enable hardwire handshake mode.*
1	X	Ignored.
2	0	Establish monitor mode 0 (all bytes displayed on terminal as they are parsed from the buffer).
	1	Establish monitor mode 1 (all bytes displayed as they are received).
3	0	Disable monitor mode.
	1	Enable the monitor mode established by bit 2.
4	0	Enable normal mode.
	1	Enable block mode.

*When hardwire handshake is enabled, the DTR line becomes a "buffer space available" flag. The line is high when available buffer space is greater than or equal to the current block size, and is held low when available buffer space is less than the current block size. This size defaults to 80 bytes unless a different value is specified by the ESC . H or ESC . I instruction.

EXAMPLE **ESC** . @;13: will establish monitor mode 1 where all bytes are displayed on the terminal as they are received by the plotter.

The Output Buffer Space Instruction, ESC . B

DESCRIPTION The output buffer space instruction, ESC . B, outputs the plotter's available buffer space.

USES This instruction is used in a software checking handshake to interrogate the plotter regarding available buffer space.

SYNTAX ESC . B

EXPLANATION No parameters are used.

RESPONSE

<DEC> The plotter's response is a decimal number in the range 0 to 1024, and represents the number of bytes of buffer space currently available for storing graphic instructions sent from the computer.

TERM This decimal number is followed by the output terminator which defaults to carriage return, CR, or is as set by ESC . M.

The Output Extended Error Instruction, ESC . E

DESCRIPTION The output extended error instruction, ESC . E, outputs a number which defines any RS-232-C related I/O error and turns off the front-panel **ERROR** light, if no HP-GL instruction errors are present.

USES The instruction is used to define what type of RS-232-C related I/O error has occurred, if any.

SYNTAX ESC . E

EXPLANATION No parameters are used.

RESPONSE

<DEC> The plotter's response is a decimal number, either 0 or in the range 10–16, followed by the output terminator. The meaning of the response is as defined in the following table.

Error No.	Meaning
0	No I/O error has occurred
10	Output instruction received while another output instruction is executing. The original instruction will continue normally; the one in error will be ignored.
11	Invalid byte received after first two characters, ESC ., in a device control instruction.
12	Invalid byte received while parsing a device control instruction. The parameter containing the invalid byte and all following parameters are defaulted.
13	Parameter out of range.
14	Too many parameters received. Additional parameters beyond the proper number are ignored; parsing of the instruction ends when a colon (normal exit) or the first byte of another instruction is received (abnormal exit).
15	A framing error, parity error, or overrun error has been detected.
16	The input buffer has overflowed. As a result, one or more bytes of data have been lost, and therefore an HP-GL error will probably occur.

NOTE: The receipt of something other than another parameter, a semicolon, or a colon will result in error 12 overwriting error 14. ■

TERM The terminator defaults to carriage return, CR, unless it is set by an ESC . M.

To check for transmission errors in a data block, first enter block mode by setting bit 4 of the second parameter of the ESC . @ instruction to logic state 1 (decimal value 16). Then begin sending data blocks, following each with the ESC . E instructiion.

In block mode, there are two possible types of response to the ESC . E instruction. If the response to the ESC . E instruction is zero, then there have been no transmission errors since the last ESC . E. In this case, the block of HP-GL instructions is transferred to the execution buffer and the instructions are executed in order. If the error number in response to the ESC . E instruction is 10–16, then there has been a transmission error since the last ESC . E. In this case, the block of

HP-GL instructions is discarded. The controller must then retransmit this block of instructions.

The following diagram illustrates block checking:

Block Checking

Computer	Plotter	Comments
ESC . E →		Any I/O errors?
	← 0<term>	No errors At this point, the plotter transfers previously-received block to the execution buffer.
Data block A →		Send a block of data Assume a byte gets garbled (bad parity).
ESC . E →		Any I/O errors?
	← 15<term>	Parity, framing, or over-run error At this point, the plotter discards the block.
Data block A → ESC . E →		Retransmit the block Any I/O errors?
	← 0<term>	No errors Plotter transfers block to the execution buffer.
Data block B		Send a block of data Assume a handshake byte gets lost, and buffer overflows.
ESC . E →		Any I/O errors?
	← 16<term>	Buffer overflow Block is discarded.
Data block B → ESC . E →		Retransmit the block Any I/O errors?
	← 0<term>	No errors Block is transferred to the execution buffer.

The Set Handshake Mode 1 Instruction, ESC . H

DESCRIPTION The set handshake mode 1 instruction, ESC . H, may be used with the enquire/acknowledge or Xon-Xoff handshake to establish parameters for the plotter's communication format.

USES It establishes the data block size, the enquiry character, and the acknowledgment string when the computer requires that the parameters set in the ESC . M instruction be used in response to the enquiry character or Xon character.

SYNTAX ESC . H [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

DEFAULT ESC . H: See ESC . I default.

EXPLANATION The two instructions, ESC . H and ESC . I, are mutually exclusive. The parameter descriptions are the same for both instructions and are given under the ESC . I instruction.

Handshake mode 1, established by this instruction, uses defaulted or specified parameters of the ESC . M and ESC . N instructions when responding to the enquiry or Xon trigger character.

The parameters used with handshake mode 1, handshake mode 2, and output responses are shown in the following table. Choose the mode and use handshake mode 1 (ESC . H) or handshake mode 2 (ESC . I) depending on the requirements of your system.

Parameter Usage in Plotter/Computer Communication

Parameter	With Handshake Characters		With Plotter Output Instructions
	In Mode 1	In Mode 2	
turnaround delay	yes	yes	yes
output trigger character	yes	no	yes
echo terminator	yes	no	yes
output terminator	yes	no	yes
output initiator*	no	no	yes
intercharacter delay	yes	yes	yes

*If an output initiator is required on enquiry character responses, it should be specified as the first character of the acknowledgment string and/or the immediate response string, depending on the system.

EXAMPLES See ESC . I and ESC . N.

The Set Handshake Mode 2 Instruction, ESC . I

DESCRIPTION The set handshake mode 2 instruction, ESC . I, may be used with the enquire/acknowledge or Xon-Xoff handshake to establish parameters for the plotter's communication format.

USES It establishes the data block size, the enquiry character, and the acknowledgment string for the enquire/acknowledge handshake when the computer expects only the turnaround delay, and not the other parameters set by ESC . M, to be included in the response to the enquiry character. It sets the Xoff threshold level and the Xon trigger character for Xon-Xoff handshake.

SYNTAX ESC . I [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

DEFAULT ESC . I : (or ESC . H :) Neither Xon-Xoff nor enquire/acknowledge handshake is enabled. Block size is 80 bytes, and there is no enquiry character or acknowledgment string. If, however, the computer is configured to send an ENQ anytime it is ready to send data to

SYNTAX **ESC** . J

EXPLANATION This instruction aborts any single device control instruction that may be partially decoded or executed. Unspecified parameters of aborted instructions are defaulted. All pending or partially transmitted output requests, from either HP-GL or device control instructions, are immediately terminated, including output responses and handshake parameters. Intermediate output operations such as turnaround delay and echo suppression are aborted, and the buffer input is enabled. The handshake and output mode parameters remain as specified.

The Abort Graphic Instruction, ESC . K

DESCRIPTION The abort graphic instruction, ESC . K, aborts any partially decoded HP-GL instruction and discards instructions in the buffer.

USES The instruction can be used as part of an initialization sequence when starting a new program or to terminate plotting of HP-GL instructions in the buffer.

SYNTAX **ESC** . K

EXPLANATION Any partially decoded HP-GL instruction is aborted and all instructions in the buffer are discarded. A partially executed instruction is allowed to finish.

The Output Buffer Size Instruction, ESC . L

DESCRIPTION The output buffer size instruction, ESC . L, outputs the size, in bytes, of the plotter's buffer.

USES The instruction is used to obtain information on the size of the plotter's buffer. This information might be used to determine parameters of instructions which set up handshaking.

SYNTAX **ESC** . L

EXPLANATION No parameters are used. The instruction causes the 7475 to output, in ASCII, a decimal number equal to the number of bytes in the plotter's buffer.

RESPONSE

<DEC>	1024
TERM	Defaults to carriage return, CR, or is as set by ESC . M.

NOTE: It is not advisable to use the ESC . L instruction when in block mode as it may disrupt communication. If you use an ESC . L instruction while in block mode, use it immediately after an ESC . E instruction.

Send the ESC . E instruction, read the response, send the ESC . L instruction, read the response, and then send the additional HP-GL instructions. ■

The Set Output Mode Instruction, ESC . M

DESCRIPTION The set output mode instruction, ESC . M, establishes parameters for the plotter's communication format.

USES The instruction is used to establish a turnaround delay, an output trigger character, an echo terminate character, and an output initiator character. It is also used to change the output terminator from its default value, carriage return.

SYNTAX ESC . M [(<DEC>) ; (<ASC>) ; (<ASC>) ; (<ASC>(; (<ASC>)) ; (<ASC>)] :

DEFAULT ESC . M : Sets the carriage return character (decimal equivalent 13) as the output terminator. It also specifies that there is no turnaround delay and no output trigger, echo terminate, or output initiator character .

EXPLANATION A colon must be used following the last parameter (if any). Use of the instruction without parameters is equivalent to ESC . M : (see DEFAULT).

A description of the instruction's parameters follows.

<DEC> The first parameter is optional. If present, it is the **turnaround delay.** The delay implemented is ((parameter × 1.1875)mod 65 536)/1.2 milliseconds. The parameter range is 0 to 54 612 milliseconds. If parameters follow, the semicolon must be included even if this decimal parameter is omitted.

<ASC> The second parameter is also optional and, if omitted, assumes its default value of 0 (no trigger character). If included, it specifies a single character which becomes the **output trigger character.** The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. If parameters follow, the semicolon must always be included, even when this parameter is omitted.

<ASC> The third parameter is optional and, if omitted, assumes its default value 0 (no echo terminate character). If included, it specifies a single character which becomes the **echo terminate character.** The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. If parameters follow, the semicolon must always be included, even when this parameter is omitted.

<ASC> . . . <ASC> The fourth parameter is optional and defaults to 13, the decimal equivalent of the single ASCII character, carriage return.

If included, the parameter may be the decimal equivalent(s) of one or two ASCII characters in the range 0 to 127. This becomes the **output terminator.** The value 0 is not transmitted and will terminate the string. If a parameter follows, the semicolon must always be included, even when this parameter is omitted. If the fifth parameter is specified, this fourth parameter must consist of two characters, or the second character must be specified as null using the semicolon.

<ASC> The fifth parameter is optional and, if omitted, assumes its default value 0 (no output initiator character). If included, it is the decimal equivalent of a single character which becomes the **output initiator** character. The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. The parameter is followed by a colon.

EXAMPLES See the ESC . N instruction.

The flowchart on the next page depicts plotter output.

The Set Extended Output and Handshake Mode Instruction, ESC . N

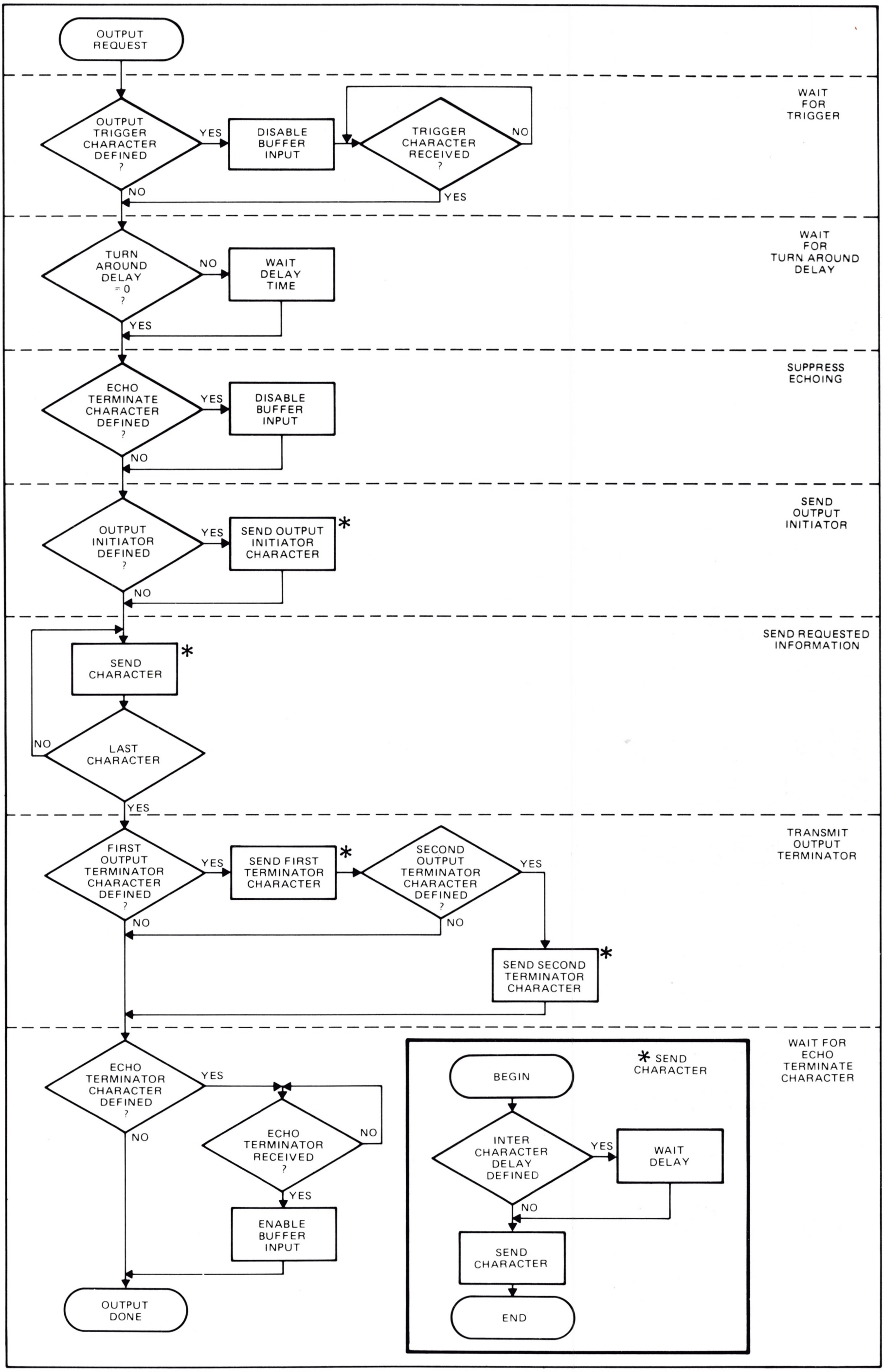
DESCRIPTION The set extended output and handshake mode instruction, ESC . N, establishes parameters for the plotter's communication format.

USES The instruction is used to specify an intercharacter delay in all handshake modes, the immediate response string for enquire/acknowledge handshake, or the Xoff trigger character(s) for the Xon-Xoff handshake.

SYNTAX ESC . N [(<DEC>) ; (<ASC>(; . . . <ASC>))] :

DEFAULT ESC . N : No intercharacter delay and no Xoff trigger character or immediate response string.

EXPLANATION A colon must be used following the last parameter. Use of the instruction without parameters is equivalent to ESC . N : (see DEFAULT).



Output Request Flow Chart

A description of the instruction's parameters follows:

<DEC> The first parameter is optional. If present, it is the **intercharacter delay.** The delay implemented is ((parameter × 1.1875)mod 65 536)/1.2 milliseconds. The parameter range is 0 to 54 612 milliseconds. If parameters follow, the semicolon must be included, even if this decimal parameter is omitted.

<ASC> . . . <ASC> This parameter is optional. If present, it is a list of the decimal equivalents of 1 to 10 ASCII characters in the range 0 to 127. For **Xon-Xoff handshake mode,** it specifies the **Xoff trigger character(s).** For **enquire/acknowledge handshake mode,** it specifies the **immediate response string.** Semicolons must separate each parameter in the list.

EXAMPLES

For Xon-Xoff Handshake

ESC . N;19: Sets the Xoff trigger character to DC3. There will be no intercharacter delay, since the first parameter is defaulted to zero by the semicolon.

For Enquire/Acknowledge Handshake

The examples given here include all handshaking instructions. In addition to illustrating the use of intercharacter delays and immediate response strings set by ESC . N, they are designed to clarify the difference between handshake mode 1 and mode 2 and give some insight into why certain values are logical choices for some parameters. The first BASIC program can be used as a handshake for the Apple II™ Plus computer with the A2B0005 serial interface card installed in slot #1 and baud rate set at 2400. Note the CHR$ function is used to send the escape character.

```
10 DIM OUT$(80)
20 IN#1
30 PR#1
40 PRINT CHR$(27);".M0;63;0;13:";CHR$(27);".N5:"
50 PRINT CHR$(27);".H80;18;49:"
60 OUT$="IN;SP1;PA500,500;":GOSUB 100
.
.
.
100 PRINT CHR$(18): INPUT Z: PRINT OUT$: RETURN
```

The following parameters are set in lines 40 and 50:

turnaround delay = 0,

output trigger character = ? (decimal equivalent 63),

no echo terminate character,

output terminator = carriage return (decimal equivalent 13),

intercharacter delay = 5,

no immediate response string,

block size = 80,

enquiry character = DC2 (decimal equivalent 18), and

acknowledgment string = 1 (decimal equivalent 49).

The subroutine in line 100 controls the handshaking. It causes the following chronological action. The enquiry character, DC2, is sent asking if the plotter has room for an 80-byte block. The plotter does not send an immediate response because that has been specified as null by its omission in the ESC . N instruction. The plotter holds its response until after it receives the output trigger character, ?. The question mark is sent by the computer when it interprets the BASIC statement INPUT to prompt for the input, Z. Z is the variable into which the acknowledgment string, 1, is read. If the acknowledgment string had been specified to contain nonnumeric characters, a string variable such as Z$ would have been used instead of Z.

The plotter waits approximately five milliseconds, the intercharacter delay, before sending the 1 and between the 1 and the output terminator, carriage return. Note the carriage return parameter could have been omitted, but carriage return still would have been sent as the output terminator because that is the default value for output terminator. If ESC . I had been used instead of ESC . H, the output terminator would not have been sent after the acknowledgment string (but it would follow responses to HP-GL output instructions). The carriage return character is a logical choice, because it is expected by the computer to delineate the end of data read by the INPUT statement.

The computer is now free to send the string OUT$, which contains HP-GL instructions, to the plotter. Note the enquiry character must be sent each time data is sent to the plotter.

Another handshake which would work using ESC . I is

```
40 PRINT CHR$(27);".I80;7;33;13:"
50 PRINT CHR$(27);".M500:";CHR$(27);".N5:"
.
.
.
100 PRINT CHR$(7): INPUT Z$: PRINT OUT$: RETURN
```

The following parameters are established:

turnaround delay = 500,

no output trigger character,

no echo terminate character,

output terminator = default value, carriage return,

intercharacter delay = 5,

no immediate response string,

block size = 80,

enquiry character = bell (decimal equivalent 7), and

acknowledgment string = ! carriage return (decimal equivalent 33, 13)

Now the computer sends the Bell character as the enquiry character. The plotter waits approximately 505 milliseconds, the total of the turnaround delay and the intercharacter delay, before sending its response. During that time, the computer will send the ? due to the INPUT statement, but the plotter ignores it. The plotter response to the enquiry character is now two characters, ! followed by a carriage return. The carriage return to terminate INPUT is now part of the acknowledgment string. No output terminator, now defaulted to carriage return, is sent because handshake mode 2 is set here by ESC . I. The output terminator, carriage return, will still follow all responses to HP-GL output instructions.

The Output Extended Status Instruction, ESC . O

DESCRIPTION The output extended status instruction, ESC . O, outputs the plotter's extended status, giving information about the state of the buffer, pinch wheels, and **VIEW** button.

USES The instruction can be used to determine, from a remote location, if the plotter is ready to plot.

SYNTAX **ESC** . O

EXPLANATION No parameters are used. Unlike the HP-GL output status instruction, OS, the ESC . O instruction does not enter the buffer but is executed immediately, subject to any turnaround or intercharacter delays specified by ESC . M and ESC . N.

RESPONSE

<DEC> The response is the decimal equivalent of a 6-bit immediate status word, followed by the output terminator. The maximum value output is 40.

The extended status word bits are as defined in the following table.

Bit	State	Decimal Value	Meaning
0–2	0	0	Not used, always zeros. Reserved for plotters with paper advance.
3	0	0	Buffer is not empty.
	1	8	Buffer is empty and ready for data.
4, 5	00	0	Ready to process or processing HP-GL instructions.
	01	16	Paper loaded, **VIEW** button pressed so graphics suspended.
	10	32	Paper lever raised so graphics suspended.

Combinations of these bits allow five different responses to the ESC . O instruction.

Response	Meaning
0	Buffer is not empty and plotter is processing HP-GL instructions.
8	Buffer is empty and is ready to process or is processing HP-GL instructions.
16	Buffer is not empty and **VIEW** has been pressed.
24	Buffer is empty and **VIEW** has been pressed.
32	Buffer is not empty and paper lever and pinch wheels are raised.
40	Buffer is empty and paper lever and pinch wheels are raised.

TERM The output terminator defaults to carriage return unless it is set by ESC . M.

The Reset Handshake Instruction, ESC . R

DESCRIPTION The reset handshake instruction, ESC . R, resets all handshake parameters to their default values.

USES The instruction may be used to set the plotter's handshake responses to a known state with hardwire handshake enabled.

SYNTAX ESC . R

EXPLANATION Executing this instruction is the same as executing the following instructions without parameters: ESC . @, ESC . H, ESC . I, ESC . M, and ESC . N. Executing this instruction, however, does not reset the HP-GL graphic instructions that may have already been sent.

The following table shows the default values of parameters used to establish handshakes.

Parameter	Value
block size	80
enquiry character	0 — no handshake enable character
acknowledgment string	0 — no handshake response string
turnaround delay	0
output trigger character	0 — no trigger character
echo terminate character	0 — no echo terminate character
output terminator	13;0; — carriage return
output initiator	0 — no output initiator
intercharacter delay	0 — no delay
immediate response string	0 — no immediate response string
monitor mode	disabled
hardwire handshake (pin 20)	enabled
buffer size	1024
Xon level	512
normal data transfer mode	enable
block data transfer mode	disable

Appendix A
An HP-IB Overview

The HP Interface Bus (HP-IB) provides an interconnecting channel for data transfer between devices on the HP-IB.

The following list defines the terms and concepts used to describe HP-IB (bus) system operations.

HP-IB System Terms

1. **Addressing** — the characters sent by a controlling device specifying which device sends information on the bus and which device(s) receives the information.
2. **Byte** — a unit of information consisting of 8 binary digits (bits).
3. **Device** — any unit that is compatible with the ANSI/IEEE 488-1978 Standard.
4. **Device Dependent** — a response to information sent on the HP-IB that is characteristic of an individual device's design, and may vary from device to device.
5. **Operator** — the person that operates either the system or any device in the system.
6. **Polling** — the process typically used by a controller to locate a device that needs to interact with the controller. There are two types of polling:
   - **Serial Poll** — a method which obtains one byte of operational information about an individual device in the system. The process must be repeated for each device from which information is desired.
   - **Parallel Poll** — a method for obtaining information about a group of devices simultaneously.

Interface Bus Concepts

Devices which communicate along the interface bus can be classified into three basic categories.

1. **Talkers** — devices which send information on the bus when they have been addressed.

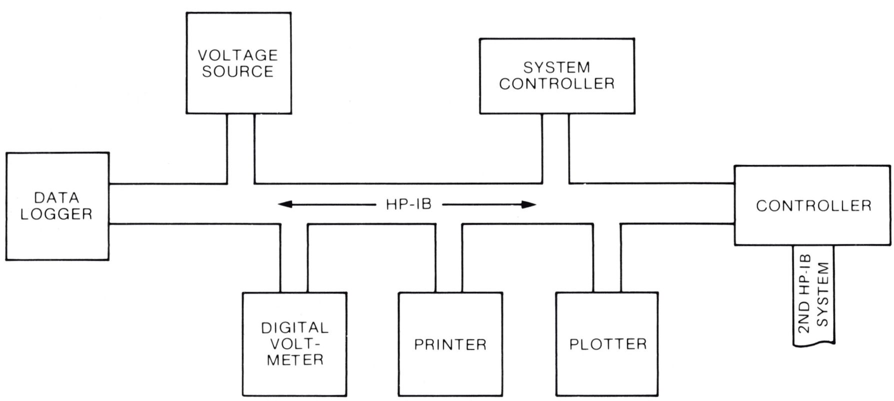
2. **Listeners —** devices which receive information sent on the bus when they have been addressed.

3. **Controllers —** devices that can specify the talker and listeners for an information transfer. Controllers can be categorized as one of two types:

   - **Active Controller —** the current controlling device on the bus. Only one device can be the active controller at any time.

   - **System Controller —** the only controller that can take priority control of the bus if it is not the current active controller. Although each bus system can have only one system controller, the system can have any number of devices capable of being the active controller.

A typical HP-IB system is shown below.



Message Concepts

Devices which communicate along the interface bus are transferring quantities of information. The transfer of information can be from one device to another device, or from one device to more than one device. These quantities of information can easily be thought of as "messages."

In turn, the messages can be classified into 12 types. The list below gives the 12 message types for the HP-IB.

1. **The Data Message.** This is the actual information which is sent from one talker to one or more listeners along the interface bus.

2. **The Trigger Message.** This message causes the listening device(s) to perform a device-dependent action when addressed.

3. **The Clear Message.** This message causes either the listening device(s) or all of the devices on the bus to return to their predefined device-dependent states.

4. **The Remote Message.** This message causes all devices currently addressed to listen to switch from local front-panel control to remote program control.
5. **The Local Message.** This message clears the Remote Message from the listening device(s) and returns the device(s) to local front-panel control.
6. **The Local Lockout Message.** This message prevents a device operator from manually inhibiting remote program control.
7. **The Clear Lockout/Local Message.** This message causes all devices on the bus to be removed from Local Lockout and revert to Local. This message also clears the Remote Message for all devices on the bus.
8. **The Require Service Message.** A device can send this message at any time to signify that the device needs some type of interaction with the controller. This message is cleared by sending the device's Status Byte Message if the device no longer requires service.
9. **The Status Byte Message.** A byte that represents the status of a single device on the bus. Bit 6 indicates whether the device sent a Require Service Message, and the remaining bits indicate operational conditions defined by the device. This byte is sent from a talking device in response to a serial poll operation performed by a controller.
10. **The Status Bit Message.** This byte represents the operational conditions of a group of devices on the bus. Each device responds on a particular bit of the byte thus identifying a device-dependent condition. This bit is typically sent by devices in response to a parallel poll operation.

    The Status Bit Message can also be used by a controller to specify the particular bit and logic level at which a device will respond when a parallel poll operation is performed. Thus, more than one device can respond on the same bit.
11. **The Pass Control Message.** This transfers the bus management responsibilities from the active controller to another controller.
12. **The Abort Message.** The system controller sends this message to unconditionally assume control of the bus from the active controller. This message terminates all bus communications (but does not implement a Clear Message).

These messages represent the full implementation of all HP-IB system capabilities. Each device in a system may be designed to use only the messages that are applicable to its purpose in the system. It is

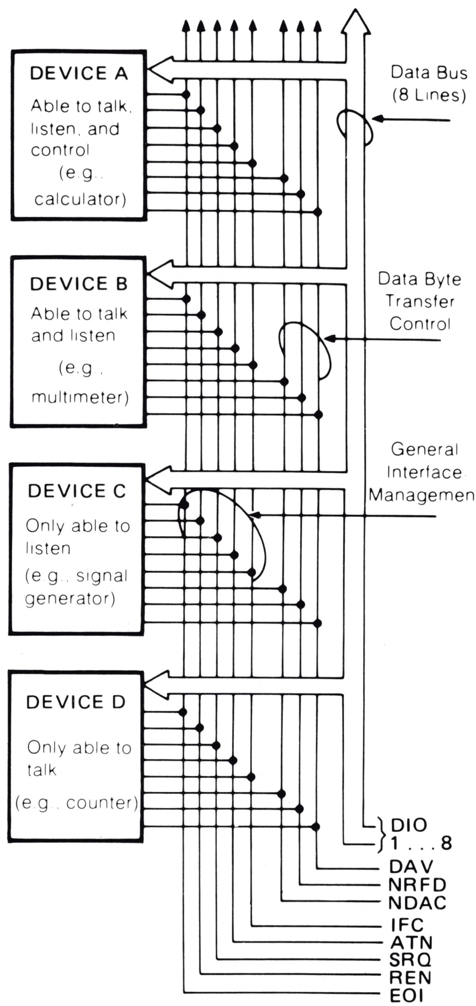
important for you to be aware of the HP-IB functions implemented on each device in your HP-IB system to ensure the operational compatibility of the system.

The HP Interface Bus

HP-IB Lines and Operations

The HP Interface Bus transfers data and commands between the components of an instrumentation system on 16 signal lines. The interface functions for each system component are performed within the component so only passive cabling is needed to connect the systems. The cables connect all instruments, controllers, and other components of the system in parallel to the signal lines.

The eight Data I/O lines (DIO1 through DIO8) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner. Data and message transfer is asynchronous, coordinated by the three handshake lines: Data Valid (DAV), Not Ready For Data (NRFD), and Not Data Accepted (NDAC). The other five lines are for management of bus activity. See the figure on the right.



HP-IB Signal Lines

Devices connected to the bus may be talkers, listeners, or controllers. The controller dictates the role of each of the other devices by setting the ATN (attention) line true and sending talk or listen addresses on the data lines. Addresses are set into each device at the time of system configuration either by switches built into the device or by jumpers on

a PC board. While the ATN line is true, all devices must listen to the data lines. When the ATN line is false, only devices that have been addressed will actively send or receive data. All others ignore the data lines.

Several listeners can be active simultaneously but only one talker can be active at a time. Whenever a talk address is put on the data lines (while ATN is true), all other talkers will be automatically unaddressed.

Information is transmitted on the data lines under sequential control of the three handshake lines (DAV, NRFD, and NDAC). No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as devices can respond, but no faster than allowed by the slowest device presently addressed as active. This permits several devices to receive the same message byte concurrently.

The ATN line is one of the five bus management lines. When ATN is true, addresses and universal commands are transmitted on only seven of the data lines using the ASCII code. When ATN is false, any code of eight bits or less understood by both talker and listener(s) may be used.

The IFC (interface clear) line places the interface system in a known quiescent state.

The REN (remote enable) line is used with the Remote, Local, and Clear Lockout/Set Local messages to select either local or remote control of each device.

Any active device can set the SRQ (service request) line true via the Require Service Message. This indicates to the controller that some device on the bus wants attention, such as a counter that has just completed a time-interval measurement and wants to transmit the reading to a printer.

The EOI (end or identify) line is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller sets both the ATN and EOI lines true, each device capable of a parallel poll indicates its current status on the DIO line assigned to it.

In the interest of cost-effectiveness, it is not necessary for every device to be capable of responding to all the lines. Each can be designed to respond only to those lines that are pertinent to its function on the bus.

The operation of the interface is generally controlled by one device equipped to act as controller. The interface transmits a group of commands to direct the other instruments on the bus in carrying out their functions of talking and listening.

The controller has two ways of sending interface messages. Multi-line messages, which cannot exist concurrently with other multi-line

messages, are sent over the eight data lines and the three handshake lines. Uni-line messages are transferred over the five individual lines of the management bus.

The commands serve several different purposes:

- Addresses or talk and listen commands select the instruments that will transmit and accept data. They are all multi-line messages.
- Universal commands cause every instrument equipped to do so to perform a specific interface operation. They include multi-line messages and three uni-line commands: interface clear (IFC), remote enable (REN), and attention (ATN).
- Addressed commands (also referred to as primary commands) are similar to universal commands, except that they affect only those devices that are addressed and are all multi-line commands. An instrument responds to an addressed command, however, only after an address has already told it to be talker or listener.
- Secondary commands are multi-line messages that are always used in series with an address, universal command, or addressed command to form a longer version of each. Thus they extend the code space when necessary.

To address an instrument, the controller uses seven of the eight data-bus lines. This allows instruments using the ASCII 7-bit code to act as controllers. As shown in the following table, five bits are available for addresses, and a total of 31 allowable addresses are available in one byte. If all secondary commands are used to extend this into a two-byte addressing capability, 961 addresses become available (31 allowable addresses in the second byte for each of the 31 allowable in the first byte.)

Command and Address Codes

Code Form	Meaning
X 0 0 A_5 A_4 A_3 A_2 A_1	Universal Commands
X 0 1 A_5 A_4 A_3 A_2 A_1 except	Listen Addresses
X 0 1 1 1 1 1 1	Unlisten Command
X 1 0 A_5 A_4 A_3 A_2 A_1 except	Talk Address
X 1 0 1 1 1 1 1	Untalk Command
X 1 1 A_5 A_4 A_3 A_2 A_1 except	Secondary Commands
X 1 1 1 1 1 1 1	Ignored

Code used when attention (ATN) is true (low).
X = don't care.

Interface Functions

Interface functions provide the physical capability to communicate via HP-IB. These functions are defined in the ANSI/IEEE 488-1978 Standard. This standard, which is the designer's guide to the bus, defines each interface function in terms of state diagrams that express all possible interactions.

Bus capability is grouped under 10 interface functions, for example: Talker, Listener, Controller, Remote/Local. The following table lists the functions, including two special cases of Controller.

HP-IB Interface Functions

Mnemonic	Interface Function Name
SH	Source Handshake
AH	Acceptor Handshake
T	Talker (or TE = Extended Talker)*
L	Listener (or LE = Extended Listener)*
SR	Service Request
RL	Remote Local
PP	Parallel Poll
DC	Device Clear
DT	Device Trigger
C	Any Controller
C_N	A Specific Controller (for example: C_A, C_B ...)
C_S	The System Controller

*Extended Talkers and Listeners use a two-byte address. Otherwise, they are the same as Talker and Listener.

Bus Messages

Since interface functions are the physical agency through which bus messages are implemented, each device must implement one or more functions to enable it to send or receive a given bus message.

The following table lists the functions required to implement each bus message. Each device's operating manual lists the functions implemented by that device. Some devices, such as the 98034A Interface, list the functions implemented directly on the device.

Functions Used by Each Bus Message

Bus Message	**Functions Required sender function → receiver function(s) (support functions)**
Data	T → L* (SH, AH)
Trigger	C → DT* (L, SH, AH)
Clear	C → DC* (L, SH, AH)
Remote	C_S → RL* (SH, AH)
Local	C → RL* (L, SH, AH)
Local Lockout	C → RL* (SH, AH)
Clear Lockout/Set Local	C_S → RL*
Require Service	SR* → C
Status Byte	T → L* (SH, AH)
Status Bit	PP* → C
Pass Control	C_A → C_B (T, SH, AH)
Abort	C_S → T, L*C

*Since more than one device can receive (or send) this message simultaneously, each device must have the function indicated by an * .

Appendix B
Instruction Syntax

HP-GL Syntax

This section lists the formal syntax for each plotter instruction in alphabetical order of the instruction's two-letter mnemonic.

Each instruction is listed with its purpose, syntax, parameter or response type, and range. If no parameter range is given, the range is -2^{15} to $2^{15}-1$. Refer to the indicated pages for details. The semicolon is included as the terminator for all instructions except the label instructions. The next mnemonic can also be used as the instruction terminator. In addition, if you have an HP-IB plotter, the line feed character can be used as a terminator. TERM means the terminator sent by the plotter at the end of output. It is CR LF in an HP-IB configuration and CR or as set by an ESC . M instruction in an RS-232-C configuration.

AA The Arc Absolute Instruction Page 3-16

AA X-coordinate,Y-coordinate,arc angle(,chord angle) ;

Purpose: Draws arc of specified number of degrees with specified smoothness; centered at X,Y coordinate, using current pen status (up or down).

Parameters: X- and Y-coordinates — integer, in plotter units unless scaling in effect; then in user units.

arc angle — integer, negative value specifies clockwise arc, positive value specifies counterclockwise arc.

chord angle — integer, defines arc smoothness in degrees. Default is 5 degrees.

AR The Arc Relative Instruction — Page 3-18

AR X-increment,Y-increment,arc angle(,chord angle) ;

Purpose: Draws arc of specified number of degrees with specified smoothness; centered relative to current pen position, using current pen status (up or down).

Parameters: X- and Y-increments — integer, in plotter units unless scaling in effect; then in user units.

arc angle — integer, negative value specifies clockwise arc, positive value specifies counterclockwise arc.

chord angle — integer, defines arc smoothness in degrees. Default is 5 degrees.

CA The Designate Alternative Character Set Instruction — Page 5-3

CA n;

Purpose: Designates the alternate character set.

Parameter: integer 0–4, 6–9, or 30–39; default set 0.

CI The Circle Instruction — Page 3-11

CI radius(,chord angle) ;

Purpose: Draws a circle of specified radius centered at current pen position.

Parameters: radius — integer, in plotter units unless scaling in effect; then in user units. Starting point at 0 degrees with positive parameter; 180 degrees with negative parameter.

chord angle — integer, defines circle smoothness in degrees. Default is 5 degrees.

CP The Character Plot Instruction — Page 5-14

CP spaces, lines ;

Purpose: Move the pen the number of spaces and lines specified.

Parameters: spaces — decimal, $\geqslant -128$ and < 128, number of CP spaces, positive value moves pen in current label direction, negative value moves pen in opposite direction.

lines — decimal, $\geqslant -128$ and < 128, number of CP lines, positive value moves pen up, negative value moves pen down in relation to current label direction.

Omitting parameters causes carriage return, line feed.

CS The Designate Standard Character Set Instruction
Page 5-3

CS m ;

Purpose: Designates the standard character set.

Parameter: integer, 0–4, 6–9 or 30–39; default set 0.

DC The Digitize Clear Instruction
Page 6-3

DC ;

Purpose: Clears digitize mode without entering a point from the front panel.

DF The Default Instruction
Page 1-11

DF ;

Purpose: Returns plotter to default conditions. See the table in Appendix C.

DI The Absolute Direction Instruction
Page 5-10

DI run, rise ;

Purpose: Sets the direction of labels.

Parameters: run, rise — decimal values, unitless. At least one must be nonzero, i.e., $|\text{parameter}| \geqslant 0.0004$.

Omitting parameters causes horizontal labels and is the same as DI 1 , 0 .

DP The Digitize Point Instruction
Page 6-2

DP ;

Purpose: Places plotter in digitize mode waiting for point to be entered from front panel.

DR The Relative Direction Instruction
Page 5-11

DR run, rise ;

Purpose: Sets the direction of labels.

Parameters: decimals, −128.0000 to +127.9999 .

run is % of $(P2_x - P1_x)$, rise is % of $(P2_y - P1_y)$.

Omitting parameters causes horizontal labels as does DR 1 , 0 .

DT The Define Terminator Instruction — Page 5-5

DT t ;

Purpose: Defines the label terminator used in LB instruction.

Parameter: ASCII character 1 to 127 except 5 and 27. Only an IN or DF instruction or use of ETX (decimal 3) as parameter restores label terminator to ETX, its default value.

EA The Edge Rectangle Absolute Instruction — Page 3-25

EA X-coordinate, Y-coordinate ;

Purpose: Draws the edge of a rectangle in absolute coordinates.

Parameters: X- and Y-coordinates

Maximum parameters — decimal, −32 768.0000 through 32 767.9999. In plotter units unless scaling in effect; then in user units. When scaling is off, parameters truncated to integers.

ER The Edge Rectangle Relative Instruction — Page 3-28

ER X-increment, Y-increment ;

Purpose: Draws the edge of a rectangle using relative coordinates.

Parameters: X-increment, Y-increment ;

Maximum parameters — decimal, −32 768.0000 through 32 767.9999. In plotter units unless scaling in effect; then in user units. When scaling is off, parameters truncated to integers.

EW The Edge Wedge Instruction — Page 3-34

EW radius, start angle, sweep angle(,chord angle) ;

Purpose: Draws the edge of a wedge.

Parameters:

Parameter	Type	Range	Default
radius	integer/ decimal	−32 768.0000– +32 767.9999	none
start angle	integer	MOD 360	none
sweep angle	integer	−32 768– +32 767	none
chord angle	integer	1–120	5°

radius — in plotter units unless scaling in effect; then in X-axis user units. The sign of the radius defines the zero-degree reference point for the start angle and sweep angle.

start angle — a positive start angle positions the radius CCW from the zero-degree reference point; a negative start angle positions the radius CW from the zero-degree reference point.

sweep angle — a positive sweep angle draws the arc segment CCW; a negative sweep angle draws the arc segment CW.

FT The Fill Type Instruction

Page 3-21

FT (type(,spacing(,angle)));

or

FT ;

Purpose: Selects a type of area fill for use with an RA, RR, or WG instruction.

Parameters:

Parameter	Number Type	Range	Default
fill type	integer	1–5	1
spacing	decimal	0–32 767.9999 (current units)	1% of the diagonal distance between P1 and P2
angle	integer	±45° increments from 0°	0°

A 0-degree angle will produce horizontal lines, a 90-degree angle will produce vertical lines, and a 45-degree angle will produce angular lines.

IM The Input Mask Instruction Page 1-14

IM E-mask value (, S-mask value(, P-mask value)) ;

Purpose: Set masks to specify which errors will cause the **ERROR** LED to come on and bit 5 of the status byte to be set, and to specify what conditions will cause a positive response to a serial or parallel poll in an HP-GL environment.

Parameters: integers 0 through 255. If parameters omitted, masks are set to 223 , 0 , 0 , the default values.

IN The Initialize Instruction Page 1-13

IN ;

Purpose: Sets the plotter to default conditions plus raises the pen, clears all HP-GL errors, and sets bit 3 of the output status byte to true (1).

The scaling points P1 and P2 are set as follows:

Paper Size	Scaling Points (Plotter Units)	
	$P1_x,P1_y$	$P2_x,P2_y$
A	250,596	10 250,7796
A4	603,521	10 603,7721
B	522,259	15 722,10 259
A3	170,602	15 370,10 602

IP The Input P1 and P2 Instruction Page 2-7

IP $P1_x$, $P1_y$ (, $P2_x$, $P2_y$) ;

Purpose: Sets scaling points.

Parameters: Integers in plotter units. Omitting parameters sets P1 and P2 to default values as listed above under the IN instruction.

IW The Input Window Instruction Page 2-12

IW $X_{\text{lower left}}$, $Y_{\text{lower left}}$, $X_{\text{upper right}}$, $Y_{\text{upper right}}$;

Purpose: Sets window inside which plotting can occur.

Parameters: Specify X- and Y-coordinates of lower-left and upper-right corners of the window.

Omitting parameters sets window to maximum plotting area, determined by the setting of the paper switches.

LB The Label Instruction Page 5-7

LB c . . . c t

Purpose: Draws the character string using the currently selected character set.

Parameters: c . . . c — ASCII characters which may include control characters.

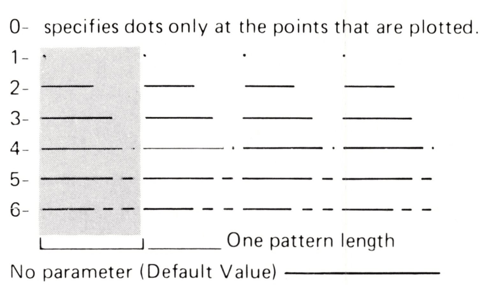
Terminator: t — label terminator defined by DT. Default is ETX, decimal 3.

LT The Line Type Instruction Page 4-6

LT pattern number (, pattern length) ;

Purpose: Sets the line type used in drawing lines.

Parameters: pattern number — integer between 0 and +6. Omitting parameter causes solid line.



pattern length — decimal, 0 to 127.9999, a percentage of diagonal distance between P1 and P2. Default 4%.

OA The Output Actual Position and Pen Status Instruction

Page 7-2

OA ;

Purpose: Used to output the pen's physical position at time of instruction.

Response: X,Y,P TERM — integers, in ASCII.

X,Y — in plotter units within current window.

P — 0, pen up or 1, pen down.

OC The Output Commanded Position and Pen Status Instruction

Page 7-3

OC ;

Purpose: Used to output the pen position and status at time of instruction.

Response: X,Y,P TERM — decimal numbers, in ASCII.

X.Y — −32 768.0000 to 32 767.9999.

P — 0, pen up or 1, pen down.

Plotter units unless scaling in effect; then in user units.

OD The Output Digitized Point and Pen Status Instruction

Page 6-3

OD ;

Purpose: Used to output the physical pen position and status for the last digitized point.

Response: X,Y,P TERM — integers, in ASCII.

X,Y — In plotter units, within mechanical limits.

P — 0, pen up or 1, pen down.

OE The Output Error Instruction

Page 7-5

OE ;

Purpose: Used to output the first HP-GL error.

Response: error number TERM — a positive ASCII integer, 0 through 8, excluding 4 and 7.

OF The Output Factors Instruction Page 7-6

OF ;

Response: 40, 40 TERM — integers, in ASCII.

OH The Output Hard-clip Limits Instruction Page 2-13

OH ;

Purpose: Used to output the lower-left and upper-right coordinates of the hard-clip limits.

Response: $X_{\text{lower left}}$, $Y_{\text{lower left}}$, $X_{\text{upper right}}$, $Y_{\text{upper right}}$, TERM — four ASCII integers in plotter units.

OI The Output Identification Instruction Page 7-6

OI ;

Purpose: Used to output the plotter's identification.

Response: 7475A TERM — ASCII string.

OO The Output Options Instruction Page 7-6

OO ;

Purpose: Used to output features implemented on the plotter.

Response: 0,1,0,0,1,0,0,0 TERM

Indicates arcs and circle instructions are included.

Indicates pen select capability is included.

OP The Output P1 and P2 Instruction

Page 2-8

OP ;

Purpose: Used to output the plotter unit coordinates of the scaling points P1 and P2.

Response: $P1_x$, $P1_y$, $P2_x$, $P2_y$ TERM — four integers in ASCII.

Range — dependent on combination setting of paper switches.

Plotting Ranges

Paper Size	Plotting Range	
	X-axis	Y-axis
A	$0 \leqslant X \leqslant 10\,365$	$0 \leqslant Y \leqslant 7962$
B	$0 \leqslant X \leqslant 16\,640$	$0 \leqslant Y \leqslant 10\,365$
A4	$0 \leqslant X \leqslant 11\,040$	$0 \leqslant Y \leqslant 7721$
A3	$0 \leqslant X \leqslant 16\,158$	$0 \leqslant X \leqslant 11\,040$

OS The Output Status Instruction

Page 7-7

OS ;

Purpose: Used to output the plotter's status.

Response: status TERM — integer in ASCII in the range 0 to 255. Power-on status, 24.

OW The Output Window Instruction

Page 2-13

OW ;

Purpose: Used to output the plotter unit coordinates of the lower-left and upper-right corners of the current window.

Response: $X_{\text{lower left}}$, $Y_{\text{lower left}}$, $X_{\text{upper right}}$, $Y_{\text{upper right}}$ TERM — integers in ASCII. Range same as OP.

PA The Plot Absolute Instruction
Page 3-4

PA X_1 coordinate, Y_1 coordinate (X_2 coordinate, Y_2 coordinate, . . ., . . ., X_n coordinate, Y_n coordinate) ;

or

PA ;

Purpose: Plots to the X,Y coordinates in the order listed using the current pen up/down status. PA ; sets absolute plotting.

Parameters: Pairs of integers representing plotter units if scaling not in effect, otherwise user units, integers or decimals.

PD The Pen Down Instruction
Page 3-2

PD ;

or

PD X_1 coordinate, Y_1 coordinate (, . . . X_n, Y_n coordinates) ;

Purpose: Programmatically lowers the pen. Parameters may be included as in PA or PR.

PR The Plot Relative Instruction
Page 3-8

PR X_1 increment, Y_1 increment (, X_2 increment, Y_2 increment, . . ., . . ., X_n increment, Y_n increment) ;

or

PR ;

Purpose: Plots, in order, to the points indicated by the X,Y increments, relative to the previous pen position. PR ; sets relative plotting for PU or PD with parameters.

Parameters: Pairs of integers representing plotter units if scaling is not in effect, otherwise user units, integers, or decimals.

PS The Paper Size Instruction
Page 1-16

PS paper size;

Purpose: Can be used to toggle between A and B, or A3 and A4 paper sizes.

Parameters: 0–3 or 4–127; 0–3 selects either B or A3 size paper; 4–127 selects A or A4 size paper.

PT The Pen Thickness Instruction — Page 3-22

PT (pen thickness) ;

or

PT ;

Purpose: Determines the spacing between the lines drawn in a solid fill.

Parameters: Decimal between 0.1 mm–5.0 mm. If parameter is omitted, defaults to .3 mm size.

PU The Pen Up Instruction — Page 3-2

PU ;

or

PU X_1 coordinate, Y_1 coordinate (, . . . X_n, Y_n coordinates) ;

Purpose: Programmatically raises the pen. Parameters may be included as in PA or PR.

RA The Shade Rectangle Absolute Instruction — Page 3-23

RA X-coordinate, Y-coordinate ;

Purpose: Defines and shades a rectangle using absolute coordinates.

Parameters: X- and Y-coordinates

Maximum parameters — decimal, −32 768.0000 through 32 767.9999. In plotter units unless scaling in effect; then in user units. When scaling is off, parameters truncated to integers.

RO The Rotate Coordinate System Instruction — Page 2-14

RO (angle in degrees) ;

or

RO ;

Purpose: Rotates the coordinate system 90 degrees.

Parameters: 0 or 90; 0 or omitting parameters turns off rotation; 90 rotates coordinate system 90 degrees.

RR The Shade Rectangle Relative Instruction

Page 3-26

RR X-increment, Y-increment ;

Purpose: Defines and shades a rectangle using relative coordinates.

Parameters: X-increment, Y-increment ;

Maximum parameters — decimal, −32 768.0000 through 32 767.9999. In plotter units unless scaling in effect; then in user units. When scaling is off, parameters truncated to integers.

SA The Select Alternate Character Set Instruction

Page 5-4

SA ;

Purpose: Selects the alternate character set designated by the CA instruction as the character set to be used for subsequent labeling.

SC The Scale Instruction

Page 2-9

SC X_{min}, X_{max}, Y_{min}, Y_{max} ;

Purpose: Scales the plotting area into user units.

Parameters: Integers.

SI The Absolute Character Size Instruction

Page 5-16

SI width, height ;

Purpose: Sets character width and height in centimetres for labels.

Parameters: width, height — decimals representing centimetres, −128.0000 to +127.9999 .

An SI instruction with no parameters will default to the following parameters based on the paper size:

Paper Size	Width	Height
A/A4	.187 cm	.269 cm
B/A3	.285 cm	.375 cm

SL The Character Slant Instruction Page 5-18

SL tan θ ;

Purpose: Establishes the slant for labeled characters.

Parameters: decimal, −128.0000 to +127.9999, interpreted as the tangent of the angle from vertical.

Omitting parameters establishes no slant, the same as the default or SL 0.

SM The Symbol Mode Instruction Page 4-4

SM character ;

Purpose: Causes specified symbol to be drawn at each plotted point.

Parameter: Any printing character ASCII 33 through 126 excluding semicolon (ASCII 59). SM space, SM control character, or SM ; cancels symbol mode.

SP The Pen Select Instruction Page 3-3

SP pen number ;

Purpose: Selects or stores a pen.

Parameter: integers. Omitting parameters or a parameter of 0 stores the pen.

SR The Relative Character Size Instruction Page 5-17

SR width, height ;

Purpose: Sets the character width and height relative to P1 and P2 for labels.

Parameters: decimals representing a percentage of vertical or horizontal distance between P1 and P2.

Width — percentage of $(P2_x - P1_x)$.

Height — percentage of $(P2_y - P1_y)$.

Omitting parameters results in value 0.75 for width and 1.5 for height.

SS The Select Standard Character Set Instruction Page 5-4

SS ;

Purpose: Selects the standard character set designated by the CS instruction as the character set used for subsequent labeling.

TL The Tick Length Instruction Page 4-2

TL tp (, tn) ;

Purpose: Establishes the length of ticks drawn with the instructions XT and YT.

Parameters: decimals.

tp — percentage of ($P2_y - P1_y$) for XT or ($P2_x - P1_x$) for YT. Denotes portion above the X-axis or to the right of the Y-axis when difference is positive.

tn — same as tp except denotes portion below the X-axis and to the left of the Y-axis.

Omitting parameters causes tick lengths tp and tn 0.5% of ($P2_y - P1_y$) or ($P2_x - P1_x$), the same as the default values.

UC The User Defined Character Instruction Page 5-19

UC (pen control ,) X-increment, Y-increment (, . . .) (, pen control) (, . . .) ;

Purpose: Draws characters or symbols defined by user.

Parameters: pen control — $\geqslant +99$ pen down or $\leqslant -99$ pen up.

X-increment, Y-increment in grid units, range, $\pm$ 98 grid units.

Omitting parameters causes a carriage return.

VS The Velocity Select Instruction Page 3-3

VS pen velocity ;

Purpose: Sets the pen velocity.

Parameters: decimal, 0 to 127.9999.

pen velocity — 1 through 38.1 interpreted as cm/s. Defaults to velocity of 38.1 cm/s, acceleration of 2 g. Any velocity parameter slows acceleration to 0.5 g.

WG The Shade Wedge Instruction Page 3-31

WG radius, start angle, sweep angle(,chord angle) ;

Purpose: Defines and fills a wedge.

Parameters:

Parameter	Type	Range	Default
radius	integer/ decimal	−32 768.0000– +32 767.9999	none
start angle	integer	MOD 360	none
sweep angle	integer	−32 768– +32 767	none
chord angle	integer	1–120	5°

radius — in plotter units unless scaling in effect; then in X-axis user units. The sign of the radius defines the zero-degree reference point for the start angle and sweep angle.

start angle — a positive start angle positions the radius CCW from the zero-degree reference point; a negative start angle positions the radius CW from the zero-degree reference point.

sweep angle — a positive sweep angle draws the arc segment CCW; a negative sweep angle draws the arc segment CW.

XT The X-Tick Instruction Page 4-2

XT ;

Purpose: Draws a vertical tick mark of the length specified by the TL instruction at the current pen position.

YT The Y-Tick Instruction Page 4-2

YT ;

Purpose: Draws a horizontal tick mark of the length specified by the TL instruction at the current pen position.

RS-232-C Instruction Syntax

This section lists the formal syntax for each RS-232-C device control instruction in alphabetical order of the escape sequence. Refer to the indicated page for details.

Plotter On

Page 10-26

ESC . (or ESC . Y

Purpose: Places the plotter in a programmed-on state.

Plotter Off

Page 10-26

ESC .) or ESC . Z

Purpose: Places the plotter in a programmed-off state.

Set Plotter Configuration

Page 10-27

ESC . @ [(<DEC>) ; (<DEC>)] :

Purpose: Enables or disables hardwire handshake mode, monitor mode, and data transmission mode.

Parameters: <DEC> — Sets maximum buffer size.

<DEC> — Data Terminal Ready (CD) line control. A decimal number in the range of 0–31.

Output Buffer Space

Page 10-28

ESC . B

Purpose: Outputs the number of byte spaces currently available for data in the buffer.

Response: <DEC> TERM — 0 to 1024.

Output Extended Error

Page 10-29

ESC . E

Purpose: Outputs a decimal code to identify the type of RS-232-C related error that occurred.

Response: <DEC> TERM — 0, no error, or 10 - 16.

Set Handshake Mode 1 Page 10-32

ESC . H [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes parameters for handshake mode 1, used when response to handshake enable character requires ESC **.** **M** parameters.

Parameters: <DEC> — Block size or Xoff threshold level.

<ASC> — Handshake enable character or not used.

<ASC> . . . <ASC> — Handshake response string of 1 to 10 characters or Xon trigger characters.

Set Handshake Mode 2 Page 10-33

ESC . I [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes parameters for handshake mode 2, used when response to handshake enable character does not require ESC **.** M parameters.

Parameters: <DEC> — Block size or Xoff threshold level.

<ASC> — Handshake enable character or omitted.

<ASC> . . . <ASC> — Handshake response string of 1 to 10 characters or Xon trigger characters.

Abort Device Control Page 10-35

ESC . J

Purpose: Aborts any partially decoded or executed device control instructions including outputs.

Abort Graphic Instruction Page 10-36

ESC . K

Purpose: Aborts any partially decoded HP-GL instruction and discards instructions in buffer.

Output Buffer Size Page 10-36

ESC . L

Purpose: Outputs the buffer size.

Response: 1024. Not output until the buffer is empty.

Set Output Mode

Page 10-37

ESC . M [(<DEC>) ; (<ASC>) ; (<ASC>) ; (<ASC>(; (<ASC>)) ; (<ASC>)] :

Purpose: Sets parameters for output.

Parameters: <DEC> — Turnaround delay, 0–54 612.

<ASC> — Output trigger character, ASCII 0–127.

<ASC> — Echo terminator character, ASCII 0–127.

<ASC> . . . <ASC> — 1 or 2 output terminators, ASCII 0–127, 0 terminates string.

<ASC> — Output initiator character, ASCII 0–127.

Set Extended Output and Handshake Mode

Page 10-38

ESC . N [(<DEC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes extended parameters for any output instruction.

Parameters: <DEC> — Delay between output characters, 0–54 612.

<ASC> . . . <ASC> — Immediate response string of 1 to 10 characters. ASCII 0–127, 0 terminates string; or Xoff trigger characters.

Output Extended Status

Page 10-42

ESC . O

Purpose: Outputs the decimal equivalent value of a 16-bit immediate status word.

Response: <DEC> TERM — a value 40 or less.

Reset Handshake

Page 10-44

ESC . R

Purpose: Resets the handshake to its default value. It is the same as sending the commands ESC . @, ESC . H, ESC . I, ESC . M, and ESC . N without parameters.

Notes

Appendix C
Reference Material

Binary Coding and Conversions

Binary is a base 2 number system using only 1's and 0's. By giving the 1's and 0's positional value, any decimal number can be represented. For example, this diagram shows how decimal 41 = binary 101001:

Decimal

$$4 \times 10^1 + 1 \times 10^0$$

$$\frac{4 \times 10 \quad + 1 \times \quad 1}{\quad 4 \qquad\qquad 1_{10}}$$

Binary

$$1 \times \; 2^5 + 0 \times \; 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$\frac{1 \times 32 \quad + 0 \times 16 \quad + 1 \times 8 \quad + 0 \times 4 \quad + 0 \times 2 \quad + 1 \times 1}{1 \qquad\quad 0 \qquad\quad 1 \qquad\quad 0 \qquad\quad 0 \qquad\quad 1_2}$$

Binary-Decimal Conversions

To convert from binary to decimal, the positional values of the 1's are added up. From the above example, this would be:

$$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$$

To convert from decimal to binary, the decimal number is divided by 2. The remainder is the binary equivalent. For example:

		Remainder (read up)	
2 ⌈41	→	1	
2 ⌈20	→	0	
2 ⌈10	→	0	= Binary 101001
2 ⌈ 5	→	1	
2 ⌈ 2	→	0	
2 ⌈ 1	→	1	

Scaling Without Using the SC Instruction

The 7475 plotter movements are in terms of plotter units where a plotter unit = 0.025 mm. While the plotter can be scaled into user units using the SC instruction, it may be convenient for you to write programs where numbers to be plotted are in some units other than plotter units. These "user units" can be converted into plotter units by the computer using the following equations:

$$X_{scaled} = \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right] A_x + P1_x - U1_x \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right]$$

$$Y_{scaled} = \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right] A_y + P1_y - U1_y \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right]$$

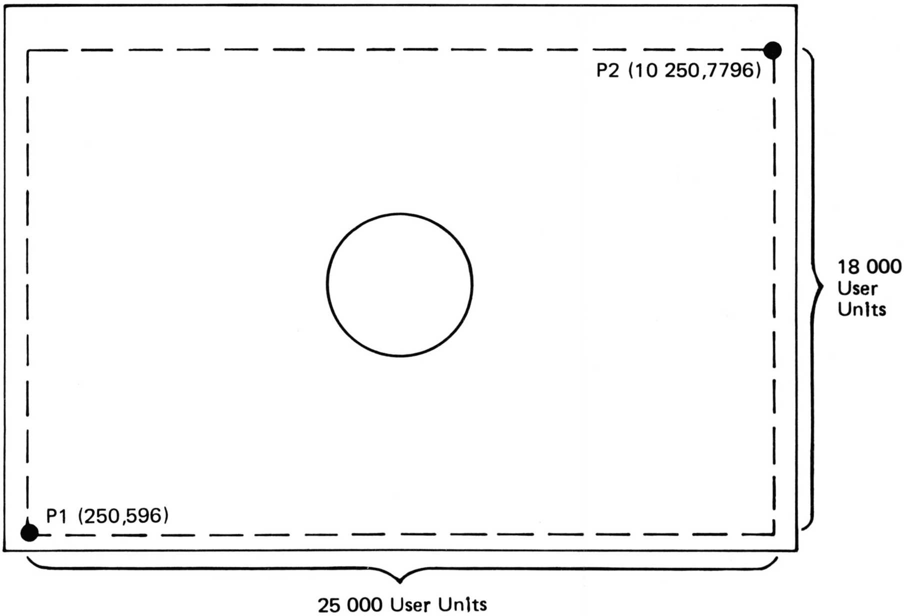
where: A_x is the X-coordinate of the desired point in user units,
A_y is the Y-coordinate of the desired point in user units,
$P1_x$ is the X-coordinate of P1 in plotter units,
$P1_y$ is the Y-coordinate of P1 in plotter units,
$P2_x$ is the X-coordinate of P2 in plotter units,
$P2_y$ is the Y-coordinate of P2 in plotter units,
$U1_x$ is the X-coordinate of P1 in user units,
$U1_y$ is the Y-coordinate of P1 in user units,
$U2_x$ is the X-coordinate of P2 in user units, and
$U2_y$ is the Y-coordinate of P2 in user units.

To demonstrate the use of the scaling equations, let's go through an example.

Example 1:

Problem

Scale the platen area (P1 = 250 , 596 and P2 = 10 250 , 7796) into user units where P1 = 0 , 0 and P2 = 25 000 , 18 000. At the center point (X = 12 500, Y = 9000), draw a circle with radius 2500 as shown on the following page.



Solution

A. Recall that the equations of a circle are:

$$X = R \cos t$$
$$Y = R \sin t$$
$$\text{where } 0 \leqslant t \leqslant 2\pi$$

B. Since we are to plot relative to a point that is not at the origin, an offset X_o, Y_o must be added to the circle equations. The offset in user units is:

$$X_o = 12\,500$$
$$Y_o = 9000$$

C. The desired circle equations are then:

$$A_x = 2500 \cos t + 12\,500$$
$$A_y = 2500 \sin t + 9000$$

D. Determine the user scale:

$$X = 0 \text{ to } 25\,000$$
$$Y = 0 \text{ to } 18\,000$$

therefore

$$U1_x = 0$$
$$U1_y = 0$$
$$U2_x = 25\,000$$
$$U2_y = 18\,000$$

E. Determine the values for P1 and P2 which were set using the IN instruction:

$$P1 = 250, 596$$
$$P2 = 10\,250, 7796$$

therefore

$$P1_x = 250$$
$$P1_y = 596$$
$$P2_x = 10\,250$$
$$P2_y = 7796$$

F. Solving for X and Y:

$$X = \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right] A_x + P1_x - U1_x \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right]$$

$$= \left[\frac{10\,250 - 250}{25\,000 - 0}\right] (2500 \cos t + 12\,500) + 250 - 0 \left[\frac{10\,250 - 250}{25\,000 - 0}\right]$$

$$= 0.4\,(2500 \cos t + 12\,500) + 250 - 0$$

$$= 1000 \cos t + 5250$$

$$Y = \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right] A_y + P1_y - U1_y \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right]$$

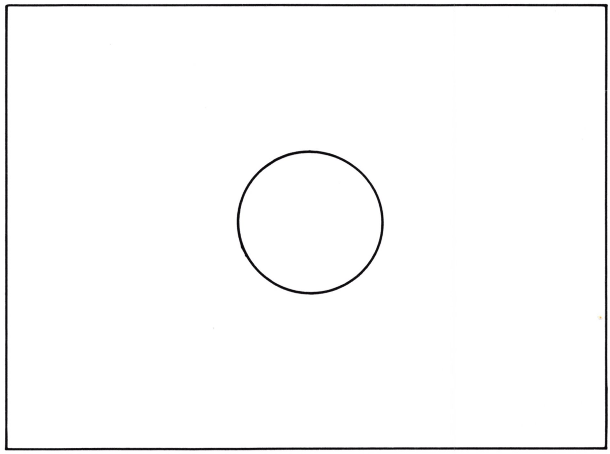
$$+ \left[\frac{7796 - 596}{18\,000 - 0}\right] (2500 \sin t + 9000) + 596 - 0 \left[\frac{7796 - 596}{18\,000 - 0}\right]$$

$$= 0.4\,(2500 \sin t + 9000) + 596 - 0$$

$$= 1000 \sin t + 4196$$

G. Sending the following program will plot the required circle using the default P1 and P2.

```
10 PRINT "IP250,596,10250,7796;SP1;"
20 FOR T=0 TO 2*PI STEP PI/20
30 X=1000*COS(T)+5250
40 Y=1000*SIN(T)+4196
50 PRINT "PA";X;Y;"PD"
60 NEXT T
70 PRINT "SP0;"
```



Plotter Default Conditions

Plotting mode	Absolute (PA)
Relative character direction	Horizontal (DR 1 , 0)
Line type	Solid line
Line pattern length	4% of the distance from P1 to P2
Fill type	Set to type 1 bidirectional solid fill
Fill spacing	1% of the diagonal distance between P1 and P2
Fill angle	Set to 0°
Input window	Mechanical limits of plotter
Relative character size	(SR .75 , 1.5) width = 0.75% of $(P2_x - P1_x)$ height = 1.5% of $(P2_y - P1_y)$
Scale	Off
Symbol mode	Off
Tick length	tp and tn = 0.5% of $(P2_x - P1_x)$ for Y-tick and 0.5% of $(P2_y - P1_y)$ for X-tick
Character set selected	Standard
Standard character set	Set 0
Alternate character set	Set 0
Label terminator	ETX (ASCII decimal equivalent 3)
Character slant	0°
Mask value	223 , 0 , 0

Digitize clear	On
Pen velocity	38.1 cm/s (15 in./s)
Pen thickness	Set to 0.3 mm
Chord angle	Set to 5 degrees for AA, AR, and CI

P1 and P2 are changed only with the initialize instruction (IN). They are not affected by device clear and the default instruction (DF).

HP-GL Error Messages

error 0 No error.

error 1 Instruction not recognized. The plotter has received an illegal character sequence.

error 2 Wrong number of parameters. Too many or too few parameters have been sent with an instruction.

error 3 Out-of-range parameters.

error 4 Not used.

error 5 Unknown character set. A character set out of the range 0–4, 6–9, 30–39 has been designated as either the standard or alternate character set.

error 6 Position overflow. An attempt to draw a character (LB or UC) or perform a CP that is located outside the plotter's numeric limit of −32 768 to +32 767.

error 7 Not used.

error 8 Vector received while pinch wheels raised.

RS-232-C Error Messages

0 No I/O error has occurred.

10 Output instruction received while another output instruction is executing. The original instruction will continue normally; the one in error will be ignored.

11 Invalid byte received after first two characters, ESC . , in a device control instruction.

12 Invalid byte received while parsing a device control instruction. The parameter containing the invalid byte and all following parameters are defaulted.

13 Parameter out of range.

14 Too many parameters received. Additional parameters beyond the proper number are ignored; parsing of the instruction ends when a colon (normal exit) or the first byte of another instruction is received (abnormal exit).

15 A framing error, parity error, or overrun error has been detected.

16 The input buffer has overflowed. As a result, one or more bytes of data have been lost, and therefore, an HP-GL error will probably occur.

The No Operation Instructions, NOP

In order to maintain software compatibility with the 9872 plotter, the 7475 recognizes six 9872-related instructions as no operation NOP instructions. These six NOP instructions are:

Automatic Pen Pickup AP	Advance Full Page AF, PG, PG1
Adaptive Velocity VA	Advance Half Page AH
Normal Velocity VN	Enable Cutter EC

If these instructions are included in a program, they are recognized by the 7475 and implemented as a NOP (i.e., they are ignored).

ASCII Character Codes

Numbers are often used as a code to represent not only values, but also alphanumeric characters such as "A" or "," or "x" or "2". One of the most common computer codes used is ASCII[1]. ASCII is an eight-bit code, containing seven data bits and one parity bit. The plotter uses ASCII for most I/O operations. No parity bit is used. For example:

Character	ASCII Binary Code	ASCII Decimal Code
A	01000001	65
B	01000010	66
?	00111111	63

[1]American Standard Code for Information Interchange.

A complete list of ASCII characters and their decimal representation and the characters drawn by the plotter in each of the 19 character sets are shown on the following pages. The 19 character sets are:

Set No.	Description	ISO Registration Number
Set 0	ANSI ASCII	006
Set 1	9825 Character Set	—
Set 2	French/German	—
Set 3	Scandinavian	—
Set 4	Spanish/Latin American	—
Set 6	JIS ASCII	014
Set 7	Roman Extensions	—
Set 8	Katakana	013
Set 9	ISO IRV (International Reference Version)	002
Set 30	ISO Swedish	010
Set 31	ISO Swedish For Names	011
Set 32	ISO Norway, Version 1	060
Set 33	ISO German	021
Set 34	ISO French	025
Set 35	ISO United Kingdom	004
Set 36	ISO Italian	015
Set 37	ISO Spanish	017
Set 38	ISO Portuguese	016
Set 39	ISO Norway, Version 2	061

7475 ASCII Code Definitions

Decimal Value	ASCII Character	All Sets
0	NULL	No Operation (NOP)
1	SOH	NOP
2	STX	NOP
3	ETX	End Label Instruction
4	ETO	NOP
5	ENQ	NOP
6	ACK	NOP
7	BEL	NOP
8	BS	Backspace
* 9	HT	Horizontal Tab (½ backspace)
10	LF	Line Feed
11	VT	Inverse Line Feed
12	FF	NOP
13	CR	Carriage Return
14	SO	Select Alternate Character Set
15	SI	Select Standard Character Set
16	DLE	NOP
17	DC1	NOP
18	DC2	NOP
19	DC3	NOP
20	DC4	NOP
21	NAK	NOP
22	SYN	NOP
23	ETB	NOP
24	CAN	NOP
25	EM	NOP
26	SUB	NOP
27	ESC	NOP
28	FS	NOP
29	GS	NOP
30	RS	NOP
31	US	NOP
32	SP	Space

*Using control character horizontal tab (decimal 9) inside a label string moves the pen one-half character space back (equivalent to a CP −.5,0). Use this tab with character set 8, Katakana, where spacing between symbols can alter the meaning of the symbol and hence the word or phrase.

NOTE: Shaded characters have the automatic backspace feature. ■

7475 ASCII Code Definitions (Continued)

Decimal Value	SET																		
	0	1	2	3	4	6	7	8	9	30	31	32	33	34	35	36	37	38	39
33	!	!	!	!	!	!	À	｡	!	!	!	!	!	!	!	!	!	!	!
34	"	"	"	"	"	"	Â	｢	"	"	"	"	"	"	"	"	"	"	"
35	#	#	£	£	¿	#	È	｣	#	#	#	#	#	£	£	£	£	#	§
36	$	$	$	$	$	$	Ê	､	¤	¤	¤	$	$	$	$	$	$	$	$
37	%	%	%	%	%	%	Ë	･	%	%	%	%	%	%	%	%	%	%	%
38	&	&	&	&	&	&	Î	ｦ	&	&	&	&	&	&	&	&	&	&	&
39	'	'	´	'	´	'	Ï	ｧ	'	'	'	'	'	'	'	'	'	'	'
40	(	(	(	(	(	(	´	ｨ	(	(	(	(	(	(	(	(	(	(	(
41	)	)	)	)	)	)	`	ｩ	)	)	)	)	)	)	)	)	)	)	)
42	*	*	*	*	*	*	^	ｪ	*	*	*	*	*	*	*	*	*	*	*
43	+	+	+	+	+	+	¨	ｫ	+	+	+	+	+	+	+	+	+	+	+
44	,	,	,	,	,	,	~	ｬ	,	,	,	,	,	,	,	,	,	,	,
45	-	-	-	-	-	-	Ù	ｭ	-	-	-	-	-	-	-	-	-	-	-
46	.	.	.	.	.	.	Û	ｮ	.	.	.	.	.	.	.	.	.	.	.
47	/	/	/	/	/	/	£	ｯ	/	/	/	/	/	/	/	/	/	/	/
48	0	0	0	0	0	0	‾	ｰ	0	0	0	0	0	0	0	0	0	0	0
49	1	1	1	1	1	1		ｱ	1	1	1	1	1	1	1	1	1	1	1
50	2	2	2	2	2	2		ｲ	2	2	2	2	2	2	2	2	2	2	2
51	3	3	3	3	3	3	•	ｳ	3	3	3	3	3	3	3	3	3	3	3
52	4	4	4	4	4	4	Ç	ｴ	4	4	4	4	4	4	4	4	4	4	4
53	5	5	5	5	5	5	ç	ｵ	5	5	5	5	5	5	5	5	5	5	5
54	6	6	6	6	6	6	Ñ	ｶ	6	6	6	6	6	6	6	6	6	6	6
55	7	7	7	7	7	7	ñ	ｷ	7	7	7	7	7	7	7	7	7	7	7
56	8	8	8	8	8	8	¡	ｸ	8	8	8	8	8	8	8	8	8	8	8
57	9	9	9	9	9	9	¿	ｹ	9	9	9	9	9	9	9	9	9	9	9
58	:	:	:	:	:	:	¤	ｺ	:	:	:	:	:	:	:	:	:	:	:
59	;	;	;	;	;	;	£	ｻ	;	;	;	;	;	;	;	;	;	;	;
60	<	<	<	<	<	<	¥	ｼ	<	<	<	<	<	<	<	<	<	<	<
61	=	=	=	=	=	=	§	ｽ	=	=	=	=	=	=	=	=	=	=	=
62	>	>	>	>	>	>	ƒ	ｾ	>	>	>	>	>	>	>	>	>	>	>
63	?	?	?	?	?	?	¢	ｿ	?	?	?	?	?	?	?	?	?	?	?
64	@	@	@	@	@	@	â	ﾀ	@	@	É	@	§	à	@	§	§	§	@

7475 ASCII Code Definitions (Continued)

Decimal Value	SET 0	1	2	3	4	6	7	8	9	30	31	32	33	34	35	36	37	38	39
65	A	A	A	A	A	A	ê	チ	A	A	A	A	A	A	A	A	A	A	A
66	B	B	B	B	B	B	ô	ツ	B	B	B	B	B	B	B	B	B	B	B
67	C	C	C	C	C	C	û	テ	C	C	C	C	C	C	C	C	C	C	C
68	D	D	D	D	D	D	á	ト	D	D	D	D	D	D	D	D	D	D	D
69	E	E	E	E	E	E	é	ナ	E	E	E	E	E	E	E	E	E	E	E
70	F	F	F	F	F	F	ó	ニ	F	F	F	F	F	F	F	F	F	F	F
71	G	G	G	G	G	G	ú	ヌ	G	G	G	G	G	G	G	G	G	G	G
72	H	H	H	H	H	H	à	ネ	H	H	H	H	H	H	H	H	H	H	H
73	I	I	I	I	I	I	è	ノ	I	I	I	I	I	I	I	I	I	I	I
74	J	J	J	J	J	J	ò	ハ	J	J	J	J	J	J	J	J	J	J	J
75	K	K	K	K	K	K	ù	ヒ	K	K	K	K	K	K	K	K	K	K	K
76	L	L	L	L	L	L	ä	フ	L	L	L	L	L	L	L	L	L	L	L
77	M	M	M	M	M	M	ë	ヘ	M	M	M	M	M	M	M	M	M	M	M
78	N	N	N	N	N	N	ö	ホ	N	N	N	N	N	N	N	N	N	N	N
79	O	O	O	O	O	O	ü	マ	O	O	O	O	O	O	O	O	O	O	O
80	P	P	P	P	P	P	Å	ミ	P	P	P	P	P	P	P	P	P	P	P
81	Q	Q	Q	Q	Q	Q	î	ム	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
82	R	R	R	R	R	R	Ø	メ	R	R	R	R	R	R	R	R	R	R	R
83	S	S	S	S	S	S	Æ	モ	S	S	S	S	S	S	S	S	S	S	S
84	T	T	T	T	T	T	å	ヤ	T	T	T	T	T	T	T	T	T	T	T
85	U	U	U	U	U	U	í	ユ	U	U	U	U	U	U	U	U	U	U	U
86	V	V	V	V	V	V	ø	ヨ	V	V	V	V	V	V	V	V	V	V	V
87	W	W	W	W	W	W	æ	ラ	W	W	W	W	W	W	W	W	W	W	W
88	X	X	X	X	X	X	Ä	リ	X	X	X	X	X	X	X	X	X	X	X
89	Y	Y	Y	Y	Y	Y	ì	ル	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
90	Z	Z	Z	Z	Z	Z	Ö	レ	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
91	[	[	[	Ø	[	[	Ü	ロ	[	Ä	Ä	Æ	Ä	°	[	°	¡	Ã	Æ
92	\	√	ç	Æ	¡	¥	É	ワ	\	Ö	Ö	Ø	Ö	ç	\	ç	Ñ	Ç	Ø
93	]	]	]	ø	]	]	ï	ン	]	Å	Å	Å	Ü	§	]	é	¿	Õ	Å
94	^	↑	^	æ	^	^	ß	゛	^	^	Ü	^	^	^	^	^	^	^	^
95	_	_	_	_	_	_	Ô	゜	_	_	_	_	_	_	_	_	_	_	_
96	`	`	`	`	`	`	Á		`	`	é	`	`	`	`	ù	`	`	`

7475 ASCII Code Definitions (Continued)

Decimal Value	SET 0	1	2	3	4	6	7	8	9	30	31	32	33	34	35	36	37	38	39
97	a	a	a	a	a	a	Ã		a	a	a	a	a	a	a	a	a	a	a
98	b	b	b	b	b	b	ã		b	b	b	b	b	b	b	b	b	b	b
99	c	c	c	c	c	c	Đ		c	c	c	c	c	c	c	c	c	c	c
100	d	d	d	d	d	d	đ		d	d	d	d	d	d	d	d	d	d	d
101	e	e	e	e	e	e	Í		e	e	e	e	e	e	e	e	e	e	e
102	f	f	f	f	f	f	Ì		f	f	f	f	f	f	f	f	f	f	f
103	g	g	g	g	g	g	Ó		g	g	g	g	g	g	g	g	g	g	g
104	h	h	h	h	h	h	Ò		h	h	h	h	h	h	h	h	h	h	h
105	i	i	i	i	i	i	Õ		i	i	i	i	i	i	i	i	i	i	i
106	j	j	j	j	j	j	õ		j	j	j	j	j	j	j	j	j	j	j
107	k	k	k	k	k	k	Š		k	k	k	k	k	k	k	k	k	k	k
108	l	l	l	l	l	l	š		l	l	l	l	l	l	l	l	l	l	l
109	m	m	m	m	m	m	Ú		m	m	m	m	m	m	m	m	m	m	m
110	n	n	n	n	n	n	Ÿ		n	n	n	n	n	n	n	n	n	n	n
111	o	o	o	o	o	o	ÿ		o	o	o	o	o	o	o	o	o	o	o
112	p	p	p	p	p	p	Þ		p	p	p	p	p	p	p	p	p	p	p
113	q	q	q	q	q	q	þ		q	q	q	q	q	q	q	q	q	q	q
114	r	r	r	r	r	r			r	r	r	r	r	r	r	r	r	r	r
115	s	s	s	s	s	s			s	s	s	s	s	s	s	s	s	s	s
116	t	t	t	t	t	t			t	t	t	t	t	t	t	t	t	t	t
117	u	u	u	u	u	u			u	u	u	u	u	u	u	u	u	u	u
118	v	v	v	v	v	v	–		v	v	v	v	v	v	v	v	v	v	v
119	w	w	w	w	w	w	¼		w	w	w	w	w	w	w	w	w	w	w
120	x	x	x	x	x	x	½		x	x	x	x	x	x	x	x	x	x	x
121	y	y	y	y	y	y	ª		y	y	y	y	y	y	y	y	y	y	y
122	z	z	z	z	z	z	º		z	z	z	z	z	z	z	z	z	z	z
123	{	₶	¨	¨	~	{	«		{	ä	ä	æ	ä	é	{	à	°	ã	æ
124	|	⊢	˙	˙	~	|	□		|	ö	ö	ø	ö	ù	|	ò	ñ	ç	ø
125	}	→	¨	¨	~	}	»		}	å	å	å	ü	è	}	è	ç	õ	å
126	~	~	'	˙	~	~	±		¯	¯	ü	¯	ß	¨	¯	ì	~	°	~

Subject Index

a

b

c

Subject Index (Continued)

Subject Index (Continued)

Subject Index (Continued)

Subject Index (Continued)

n

o

Subject Index (Continued)

p

r

Subject Index (Continued)

S

Subject Index (Continued)

t

u

Subject Index (Continued)

HEWLETT PACKARD

NOVEMBER 1986
PRINTED IN SPAIN

PART NO. 07475-90001